

Medical Equipment Supply and Maintenance

On-prem Final Project Report

Group 8

Pooja Ramesh

Sai Varun Kumar Namburi

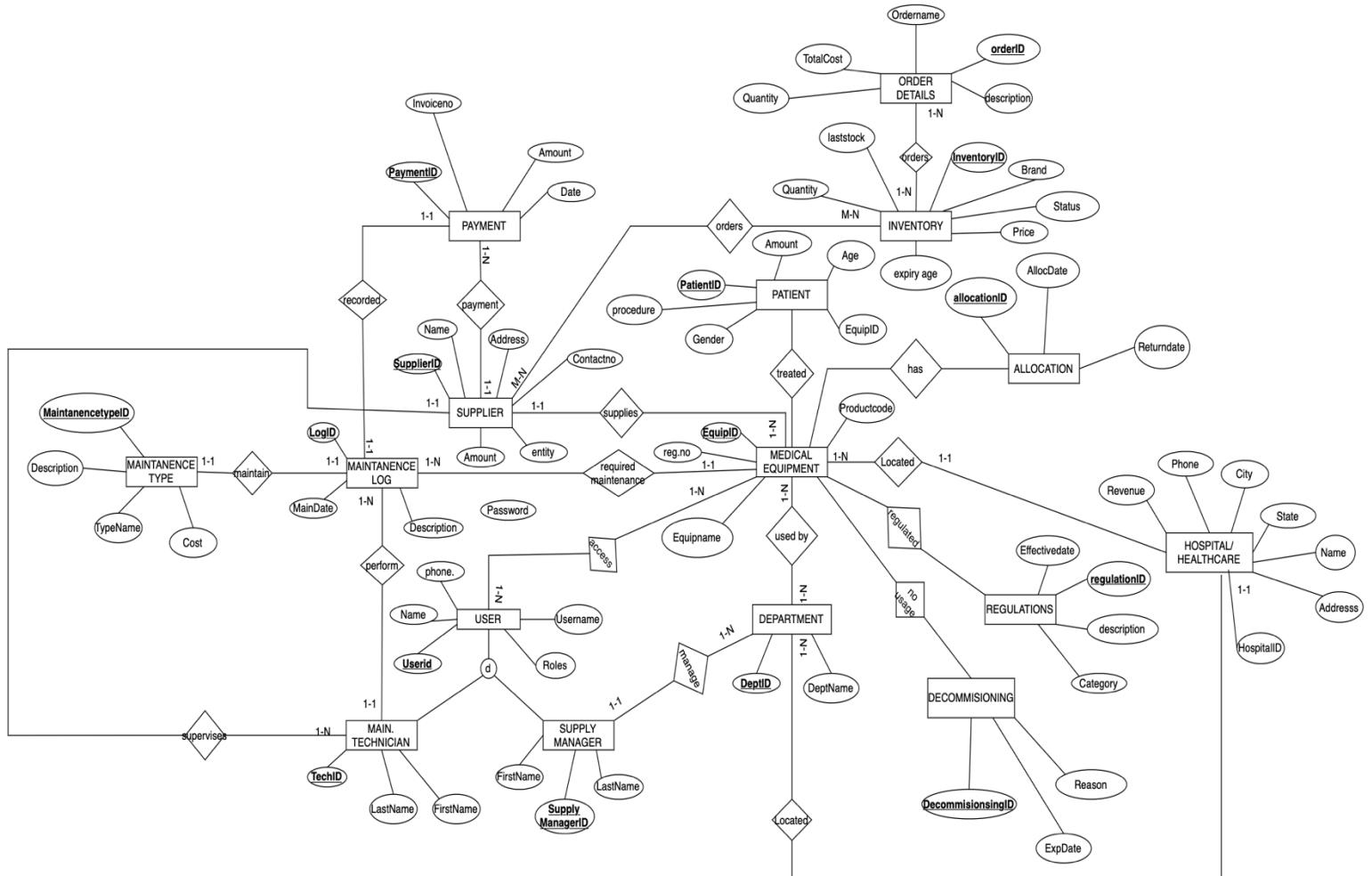
Ramesh.po@northeastern.edu

Namburi.sai@northeastern.edu

Problem Definition:

A real-world challenge faced by the healthcare domain is continuous and reliable availability of equipments. To address the comprehensive management of medical equipment, we integrate supply chain principles in healthcare facilities. To ensure that the supply and maintenance is smooth, we aim to analyze the data primarily focusing on supplier details, revenue generation, maintenance logs, hospitals/ healthcare facilities supplied to, equipment inventory, maintenance scheduling, patient information, allocation of equipments to various departments, and user access records.

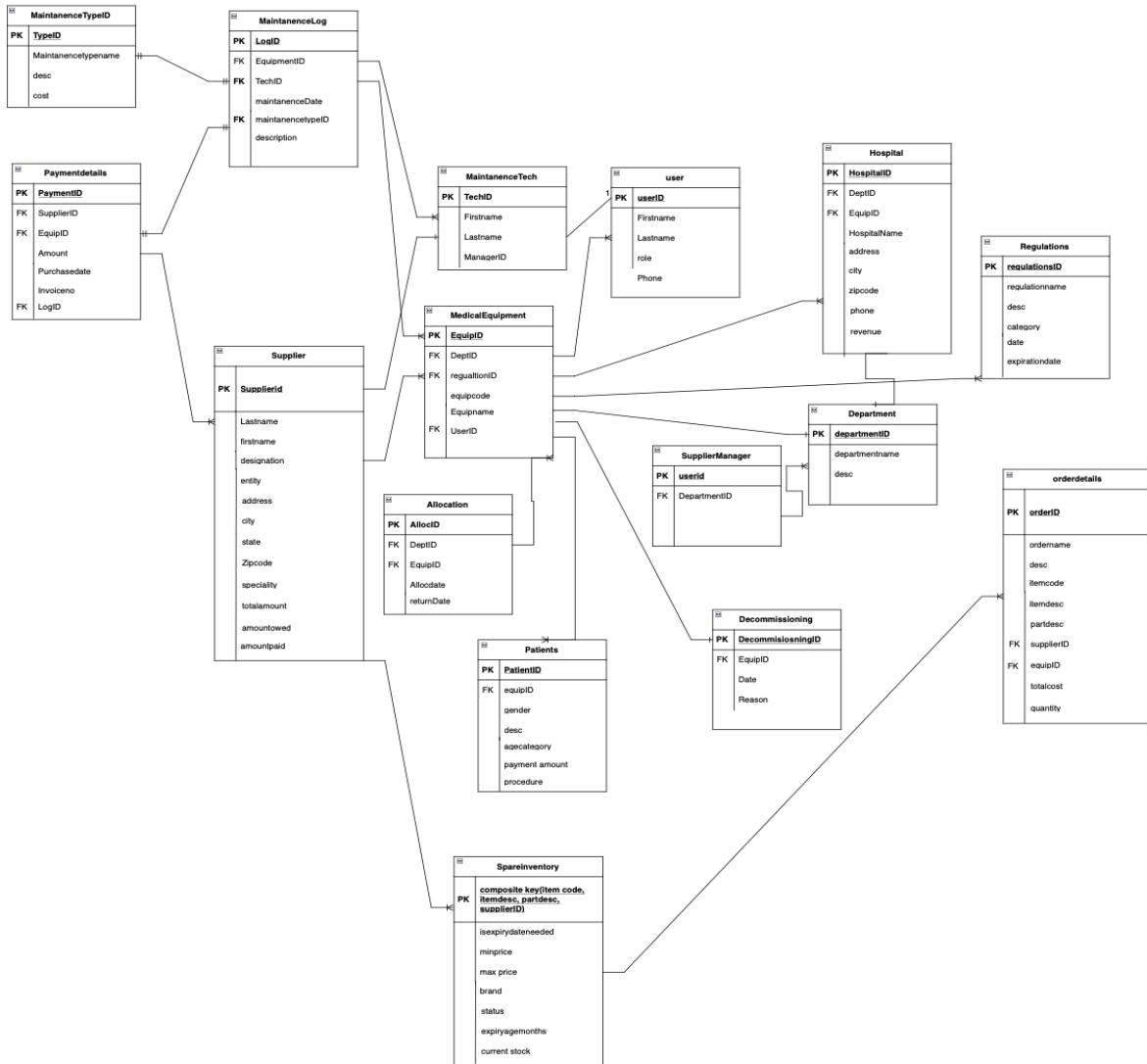
EER Diagram:



Relational Model:

Bold -> Primary Key; *Italic* -> Foreign Key

- Hospital (**HospitalID**, HospitalName, Address, City, State, Zipcode, PhoneNo, Revenue, *EquipmentID, DepartmentID*)
- Supplier (**SupplierID**, SupplierLastName, SupplierFirstName, SupplierDesignation, SupplierGender, SupplierEntity, SupplierAddress, SupplierCity, SupplierState, SupplierZipcode, SupplierSpeciality, TotalAmount, AmountPaid, AmountOwed)
- MaintenanceTechnician (**TechnicianID**, FirstName, LastName, *ManagerID*)
- Department (**DepartmentID**, DepartmentName, Description)
- Regulations (**RegulationID**, RegulationName, Description, Category, EffectiveDate, ExpirationDate)
- MedicalEquipment (**EquipmentID**, EquipmentName, EquipmentCode, *RegulationID, DepartmentID, UserID*)
- Decommissioning (**DecommissiongID**, *EquipmentID*, DecommissiongDate, Reason)
- Allocation (**AllocationID**, AllocationDate, *DeapartmentID*, ReturnDate, *EquipmentID*)
MaintenanceType (**MaintenanceTypeID**, MaintenanceTypeName, Description, MaintenanceCost)
- MaintenanceLog (**LogID**, *EquipmentID, TechnicianID*, MaintenanceDate, *MaintenanceTypeID*, Description)
- Patients (**PatientID**, Gender, AgeCategory, Description, PaymentAmount, Procedure, *EquipmentID*)
- EquipmentSupplier (*EquipmentID, SupplierID*)
- SpareInventory (ItemCode, ItemDescription, PartDescription, IsExpiryDateRequired, MinPrice, MaxPrice, Brand, Status, ExpiryAgeMonth, CurrentStock, SupplierID)
ItemCode, ItemDescription, PartDescription, SupplierID -> Composite Key
- UserAccess (**UserID**, UserFirstName, UserLastName, Role, PhoneNo)
- SupplierManager (UserID, DepartmentID)
- Orderdetails (OrderID, OrderName, Description, ItemCode, ItemDescription, PartDescription, SupplierID, EquipmentID, TotalCost, Quantity)
- PaymentDetails (**PaymentID**, *SupplierID, EquipmentID*, Amount, PurchaseDate, InvoiceNo, *LogID*);

Relational Schema:

Creation of MedicalOD schema and data population

Data population methodologies:

We used various techniques ranging from manual entry to data generation for populating the relational database. We noticed that there are very limited rows in the maintenance type table and hence, we performed insert into statements for the same. For real time transactions and patient data, we found a data simulator called syntea. For most of our tables, we used csv import to populate the data. For details like maintenance log, purchase order details, we generated and cleaned the data using python.

Tables:

The screenshot shows a database interface with a dark theme. On the left, there is a tree view of database objects. At the top level, there is a folder icon followed by 'medicalequipment_OD'. Below it is a folder icon followed by 'Schemas'. Under 'schemas', there is another folder icon followed by 'medicalod'. This is followed by a table icon followed by 'Tables'. The 'Tables' section contains a list of 18 tables, each with a small table icon and a name. To the right of each table name is its size in kilobytes (K), megabytes (M), or bytes (B). The table names and their sizes are as follows:

Table Name	Size
allocation	496K
decommissioning	64K
department	64K
equipmentsupplier	568K
hospital	136K
maintenancelog	1.5M
maintenancetechnician	512K
maintenancetype	32K
medicalequipment	816K
orderdetails	1.5M
patients	3.9M
paymentdetails	1.4M
regulations	32K
spareinventory	192K
supplier	2.5M
suppliermanager	8K
useraccess	64K

1. **Allocation Table:** This table gives the details of allocation of the equipments, allocation date and the return date assigned.

The screenshot shows a PostgreSQL database interface with two tabs: 'CreateStatements.sql' and 'Script-5'. The 'Script-5' tab contains the SQL query: 'select * from medicalod.allocation a;'. Below the query is a table titled 'allocation 1' with 12 rows of data. The columns are: allocationid, allocationdate, deapartmentid, returndate, and equipmentid. The data is as follows:

	allocationid	allocationdate	deapartmentid	returndate	equipmentid
1	AL0001	2023-07-25	1	2023-08-08	1
2	AL0002	2023-03-28	2	2023-04-08	2
3	AL0003	2023-09-22	3	2023-10-07	3
4	AL0004	2023-03-15	4	2023-03-30	4
5	AL0005	2023-07-27	5	2023-08-09	5
6	AL0006	2023-05-04	6	2023-05-17	6
7	AL0007	2023-12-24	7	2024-01-08	7
8	AL0008	2023-09-01	8	2023-09-16	8
9	AL0009	2023-04-13	9	2023-04-28	9
10	AL0010	2023-12-03	10	2023-12-15	10
11	AL0011	2023-04-02	11	2023-04-14	11
12	AL0012	2023-10-26	12	2023-11-06	12

2. **Decommissioning Table:** This outlines the discarded equipments and the corresponding reason.

The screenshot shows a PostgreSQL database interface with two tabs: 'CreateStatements.sql' and 'Script-5'. The 'Script-5' tab contains the SQL query: 'select * from medicalod.decommissioning d;'. Below the query is a table titled 'decommissioning 1' with 12 rows of data. The columns are: decommissionid, equipmentid, decommissiondate, and reason. The data is as follows:

	decommissionid	equipmentid	decommissiondate	reason
1	DC001	3,227	2023-11-14 00:00:00.000	Upgrade
2	DC002	3,622	2023-08-31 00:00:00.000	Upgrade
3	DC003	3,555	2023-11-10 00:00:00.000	Upgrade
4	DC004	5,449	2023-07-14 00:00:00.000	Safety concerns
5	DC005	2,235	2023-07-15 00:00:00.000	Equipment malfunction
6	DC006	2,847	2023-08-22 00:00:00.000	End of life
7	DC007	3,596	2023-11-25 00:00:00.000	End of life
8	DC008	1,921	2023-02-16 00:00:00.000	Upgrade
9	DC009	4,345	2023-03-14 00:00:00.000	End of life
10	DC010	997	2023-08-19 00:00:00.000	End of life
11	DC011	124	2023-03-13 00:00:00.000	Equipment malfunction
12	DC012	5,253	2023-09-24 00:00:00.000	Obsolete technology

3. Department Table:

The screenshot shows a database interface with a query window and a results grid. The query is:

```
select * from medicalod.department d ;
```

The results grid displays 10 rows of data:

departmentid	departmentname	description
1	Acupuncturist	
2	Addiction Medicine	
3	All Other Suppliers	
4	Allergy/Immunology	
5	Ambulatory Surgical Center	
6	Anesthesiologist Assistants	
7	Anesthesiology	
8	Assisted Living Facility	
9	Audiologist (billing independently)	
10	Audiologist-Hearing Aid Fitter	

4. **Hospital Table:** Contains the comprehensive list of all the hospitals/healthcare facilities along with the details and the amount they spend on supplies.

The screenshot shows a database interface with a query window and a results grid. The query is:

```
select * from medicalod.hospital h  
where state='MA';
```

The results grid displays 10 rows of data:

hospitalid	hospitalname	address	city	state	zipcode	phoneno	revenue	equipment
1	HEALTHALLIANCE HOSPITALS INC	60 HOSPITAL ROAD	LEOMINSTER	MA	1453	9784662000	198,002.28	
2	MOUNT AUBURN HOSPITAL	330 MOUNT AUBURN STREET	CAMBRIDGE	MA	2138	6174923500	288,569.17	
3	STURDY MEMORIAL HOSPITAL	211 PARK STREET	ATTLEBORO	MA	2703	5082225200	287,355.03	
4	LAWRENCE GENERAL HOSPITAL	ONE GENERAL STREET	LAWRENCE	MA	1842	9786834000	163,386.98	
5	CAMBRIDGE HEALTH ALLIANCE	1493 CAMBRIDGE STREET	CAMBRIDGE	MA	2138	6176652300	406,053.19	
6	CAPE COD HOSPITAL	88 LEWIS BAY ROAD	HYANNIS	MA	2601	5087711800	226,779.11	
7	COOLEY DICKINSON HOSPITAL INC THE	30 LOCUST STREET	NORTHAMPTON	MA	1060	4135822000	207,766.77	
8	BAYSTATE FRANKLIN MEDICAL CENTER	164 HIGH STREET	GREENFIELD	MA	1301	4137730211	137,761.9	
9	CARNEY HOSPITAL	2100 DORCHESTER AVENUE	BOSTON	MA	2124	6175062000	47,143.4	
10	HARRINGTON MEMORIAL HOSPITAL-1	100 SOUTH STREET	SOUTHBRIDGE	MA	1550	5087659771	179,609.83	

5. Maintenance Log Table: Compiled list of all the technician logs and equipment accessed for maintenance.

The screenshot shows a PostgreSQL database interface with a query editor and a results grid. The query in the editor is:

```
select * from medicalod.maintenancelog m
where maintainedate < '2023-07-15';
```

The results grid is titled "maintenancelog 1" and contains the following data:

logid	equipmentid	technicianid	maintainedate	maintenancetypeid	description
1	4,640	4,331	2023-01-23	2	Disinfection and sterilization
2	5	1,384	2023-01-26	3	Replaced worn-out components
3	6	2,116	2023-07-13	3	Routine inspection and cleaning
4	9	3,282	2023-02-25	5	Safety inspection and compliance check
5	10	3,995	2023-05-03	6	Safety inspection and compliance check
6	12	2,859	2023-06-02	2	Verified calibration accuracy
7	13	1,725	2023-02-23	9	Calibrated equipment for accuracy
8	14	5,849	2023-02-21	6	Preventive maintenance
9	15	4,905	2023-03-10	8	Replaced worn-out components
10	18	2,018	2023-04-18	5	Replaced faulty sensor
11	19	2,339	2023-01-10	7	Performed software update
12	20	5,449	2023-04-08	1	Disinfection and sterilization
13	21	2,185	2023-03-11	2	Calibrated equipment for accuracy

6. Maintenance Technician Table: List of technicians and their managers who provide the service.

The screenshot shows a PostgreSQL database interface with a query editor and a results grid. The query in the editor is:

```
select * from medicalod.maintenancetechnician m
where managerid > 30;
```

The results grid is titled "maintenancetechnician 1" and contains the following data:

technicianid	firstname	lastname	managerid
31	Herbert	Garfield	31
32	Ronald	Wilson	32
33	Millard	Lincoln	33
34	George	McKinley	34
35	Richard	Johnson	35
36	Grover	Carter	36
37	Benjamin	Nixon	37
38	Rutherford	Jefferson	38
39	Warren	Fillmore	39
40	Chester	Jefferson	40

IE7374 Data Warehousing and Data Integration

7. Maintenance Type Table: The various types of maintenance that a medical equipment undergoes.

A screenshot of a PostgreSQL database interface. The query window contains the SQL command: `select * from medicalod.maintenancetype m`. The results are displayed in a grid titled "maintenancetype 1". The grid has columns: maintenancetypeid, maintenancetypename, description, and maintenancecost. The data shows ten rows of maintenance types with their descriptions and costs.

Grid	1	maintenancetypename	description	123
Text	1	Preventive Maintenance	Routine maintenance to prevent issues proactively.	500
Text	2	Corrective Maintenance	Repair or fix equipment when it malfunctions.	750
Text	3	Scheduled Maintenance	Planned maintenance based on a fixed schedule.	1,200
Text	4	Emergency Maintenance	Immediate maintenance for critical issues.	980
Text	5	Calibration	Adjusting and calibrating equipment for accuracy.	620
Text	6	Software Update	Updating equipment software for performance and security.	890
Text	7	Cleaning and Sanitization	Cleaning and sanitizing equipment regularly.	1,100
Text	8	Hardware Replacement	Replacing hardware components as needed.	950
Text	9	Inspection	Regular inspections to identify potential issues.	1,350
Text	10	Upgrades	Upgrading equipment for improved functionality.	720

8. Medical Equipment Table:

A screenshot of a PostgreSQL database interface. The query window contains the SQL command: `select * from medicalod.medicalequipment m ;`. The results are displayed in a grid titled "medicalequipment 1". The grid has columns: equipmentid, equipmentname, equipmentcode, regulationid, departmentid, userid. The data shows 14 rows of medical equipment items with their names, codes, and associated users.

Grid	1	equipmentid	equipmentname	equipmentcode	regulationid	departmentid	userid
Text	1	1	Protector, Dental	BRW	R001	1	1
Text	2	2	Stool, Anesthesia	BRX	R002	2	2
Text	3	3	Cabinet, Table And Tray, Anesthesia	BRY	R003	3	3
Text	4	4	Analyzer, Gas, Helium, Gaseous-Phase	BSE	R004	4	4
Text	5	5	Absorber, Carbon-Dioxide	BSF	R005	5	5
Text	6	6	Algesimeter, Powered	BSI	R006	6	6
Text	7	7	Mask, Gas, Anesthetic	BSJ	R007	7	7
Text	8	8	Cuff, Tracheal Tube, Inflatable	BSK	R008	8	8
Text	9	9	Filter, Conduction, Anesthetic	BSN	R009	9	9
Text	10	10	Catheter, Conduction, Anesthetic	BSO	R010	10	10
Text	11	11	Needle, Conduction, Anesthetic (W/Wo Introducer)	BSP	R011	11	11
Text	12	12	Stylet, Tracheal Tube	BSR	R012	12	12
Text	13	13	Catheters, Suction, Tracheobronchial	BSY	R013	13	13
Text	14	14	Gas-Machine, Anesthesia	BSZ	R014	14	14

9. Order Details Table: List of orders, along with the item, product codes and the suppliers.

A screenshot of a PostgreSQL database interface. The query window contains the SQL command: `select * from medicalod.orderdetails o where itemdescription = 'Bulbs/ Lamps';`. The results are displayed in a grid titled "orderdetails 1". The grid has columns: orderid, ordername, description, itemcode, itemdescription. The data shows 15 rows of order details for items related to 'Bulbs/ Lamps'.

Grid	1	orderid	ordername	description	itemcode	itemdescription
Text	1	7	Monitor, Oxygen, Cutaneous, For Uses Other Than For Infant Not	Description for Monitor, Oxygen, Cutaneous, For Uses Other Than	BP0003	Bulbs/ Lamps
Text	2	10	Non-Continuous Ventilator For Emergency Use	Description for Non-Continuous Ventilator For Emergency Use	BP0003	Bulbs/ Lamps
Text	3	16	Monitor, Oxygen, Cutaneous, For Uses Other Than For Infant Not	Description for Monitor, Oxygen, Cutaneous, For Uses Other Than	BP0003	Bulbs/ Lamps
Text	4	17	Oximeter	Description for Oximeter	BP0003	Bulbs/ Lamps
Text	5	26	Mask, Oxygen, Non-Rebreathing	Description for Mask, Oxygen, Non-Rebreathing	BP0003	Bulbs/ Lamps
Text	6	29	Analyzer, Gas, Isoflurane, Gaseous-Phase (Anesthetic Concentric	Description for Analyzer, Gas, Isoflurane, Gaseous-Phase (Anesthe	BP0003	Bulbs/ Lamps
Text	7	30	Needle, Spinal, Short Term, Reprocessed	Description for Needle, Spinal, Short Term, Reprocessed	BP0003	Bulbs/ Lamps
Text	8	31	Bed, Rocking, Breathing Assist Nebulizer, Medicinal, Non-Ventilat	Description for Bed, Rocking, Breathing Assist Nebulizer, Medicin	BP0003	Bulbs/ Lamps
Text	9	37	Device, Rebreathing Tubing, Pressure And Accessories	Description for Device, Rebreathing Tubing, Pressure And Accesso	BP0003	Bulbs/ Lamps
Text	10	65	Tube, Tracheal, Reprocessed	Description for Tube, Tracheal, Reprocessed	BP0003	Bulbs/ Lamps
Text	11	69	Ventilator, Continuous, Non-Life-Supporting	Description for Ventilator, Continuous, Non-Life-Supporting	BP0003	Bulbs/ Lamps
Text	12	79	Cylinder, Compressed Gas, And Valve	Description for Cylinder, Compressed Gas, And Valve	BP0003	Bulbs/ Lamps
Text	13	83	Anesthesia Kit	Description for Anesthesia Kit	BP0003	Bulbs/ Lamps
Text	14	89	Calculator, Drug Dose	Description for Calculator, Drug Dose	BP0003	Bulbs/ Lamps
Text	15	100	Protector, Dental Stool	Description for Protector, Dental Stool	BP0003	Bulbs/ Lamps

10. Patients Table:

SQL Query:

```
select * from medicalod.patients p ;
```

Table View:

	patientid	gender	agecategory	description	paymentamount	procedure	equipmentid
1	1	Male	Between 25-35 years old	Hospital bed, semi-electric (head and foot adjustment), with any type side rails,	80	9	3,236
2	2	Male	Between 55-65 years old	Hospital bed, semi-electric (head and foot adjustment), with any type side rails,	100	3	2,632
3	3	Male	Above 65 years old	Walker, folding, wheeled, adjustable or fixed height	80	4	1,231
4	4	Male	Between 35-45 years old	Seat attachment, walker	20	3	5,266
5	5	Male	Between 45-55 years old	Commode chair, mobile or stationary, with fixed arms	90	3	5,042
6	6	Male	Under 25 years old	Gel or gel-like pressure pad for mattress, standard mattress length and width	250	2	867
7	7	Male	Between 35-45 years old	Hospital bed, semi-electric (head and foot adjustment), with any type side rails,	80	20	2,791
8	8	Male	Between 55-65 years old	Hospital bed, semi-electric (head and foot adjustment), with any type side rails,	100	5	4,535
9	9	Male	Between 45-55 years old	Portable gaseous oxygen system, rental; includes portable container, regulator, f	20	6	4,846
10	10	Male	Between 55-65 years old	Nebulizer, with compressor	10	4	2,530

11. Payment Details Table: Contains the details of the supply and maintenance total costs of medical equipments from each supplier.

SQL Query:

```
select * from medicalod.paymentdetails p
```

Table View:

	123 paymentid	123 supplierid	123 equipmentid	123 amount	123 purchasedate	123 invoiceno	123 logid
1	1	1	1,899	18,894.85	2023-06-12	INV-00001	1
2		2	3,611	2,831.04	2023-06-16	INV-00002	2
3		3	5,192	14,667.99	2023-03-03	INV-00003	3
4		4	2,797	93,940.7	2023-10-01	INV-00004	4
5		5	2,948	18,335.55	2023-02-05	INV-00005	5
6		6	5,596	3,948.64	2023-10-11	INV-00006	6
7		7	3,430	56,000.97	2023-06-01	INV-00007	7
8		8	3,364	9,062.53	2023-08-25	INV-00008	8
9		9	4,334	6,507.85	2023-10-24	INV-00009	9
10		10	5,494	301,050.59	2023-06-25	INV-00010	10

12. Regulations Table: List of maintenance regulations and codes to be met and followed for medical equipment.

SQL Query:

```
select * from medicalod.regulations r ;
```

Table View:

	regulationid	regulationname	description	category	effective date	expiration date
1	R001	Medical Equipment Maintenance - 2023	Standards for medical equipment maintenance in 2023	Maintenance	2023-01-01	2023-12-31
2	R002	Supply Chain Guidelines - 2023	Guidelines for medical equipment supply in 2023	Supply	2023-03-15	2023-12-31
3	R003	Equipment Maintenance Best Practices	Best practices for medical equipment maintenance	Maintenance	2021-05-10	2024-12-31
4	R004	Procurement Regulations - 2022	Regulations for equipment procurement in 2022	Supply	2022-02-01	2023-12-31
5	R005	Maintenance Reporting Standards	Standards for reporting maintenance activities	Maintenance	2020-12-15	2025-12-31
6	R006	Equipment Supply Code of Ethics	Code of ethics for medical equipment suppliers	Supply	2019-06-20	2023-12-31
7	R007	Equipment Inspection Guidelines	Guidelines for inspecting medical equipment	Maintenance	2018-03-01	2023-12-31
8	R008	Procurement Transparency Act	Act promoting transparency in equipment procurement	Supply	2020-08-01	2024-12-31
9	R009	Equipment Calibration Standards	Standards for calibrating medical equipment	Maintenance	2019-11-15	2025-12-31
10	R010	Supply Chain Sustainability Practices	Sustainability practices for equipment supply	Supply	2021-04-05	2023-12-31

13. Supplier Table:

```
select * from medicalod.supplier s
where supplierstate = 'IL';
```

supplier 1 ×

Grid	stname	supplierfirstname	supplierdesignation	suppliergender	supplierrarity	supplieraddress	suppliercity	supplierstate	suppl
1	ERIN	M.D.	F	Individual	240 E HURON ST	CHICAGO	IL	6061129	
2	ANOOPA	MD	F	Individual	5841 S MARYLAND AVE	CHICAGO	IL	6063714	
3	HERMAN	M.D.	M	Individual	9977 WOODS DR	SKOKIE	IL	600771C	
4	DEEPAK	M.D.	M	Individual	530 NE GLEN OAK AVE	PEORIA	IL	6163700	
5	ERNESTO	MD	M	Individual	3525 W PETERSON AVE	CHICAGO	IL	6065933	
6	JILL	MD	F	Individual	6831 N PEARTREE LN	PEORIA	IL	6161524	
7	SALVA	M.B.S.	F	Individual	101 N WALNUT ST	PINCKNEYVILLE	IL	6227410	
8	TASHA	N.P.	F	Individual	2614 W JEFFERSON ST	JOLIET	IL	6043564	
9	SANDEEP	M.D.	M	Individual	4700 W 95TH ST	OAK LAWN	IL	6045325	
10	LORI	M.D.	F	Individual	500 HEALTH CENTER DR	MATTOON	IL	6193892	

14. Supplier Manager Table: Contains details of the supply chain manager for each supplier.

```
select * from medicalod.suppliermanager s ;
```

suppliermanager 1 ×

Grid	userid	departmentid
1	10	1
2	11	2
3	12	3
4	13	4
5	14	5
6	15	6
7	16	7
8	17	8
9	18	9
10	19	10
11	20	11

Warehouse Design Proposal:

Problem Definition:

Our main aim is to implement a comprehensive data solution to by creating a centralize repository for all equipment related data. This will help in ensuring proactive maintenance, cost control, inventory optimization and regulate compliance. Here, we further aim to create a scheduling table which will regularize maintenance schedules for each equipment and follow the regulation. Furthermore, category/department wise segregation of equipment will ease the process of tracking and ensure the entire system is reliable.

Dimensions:

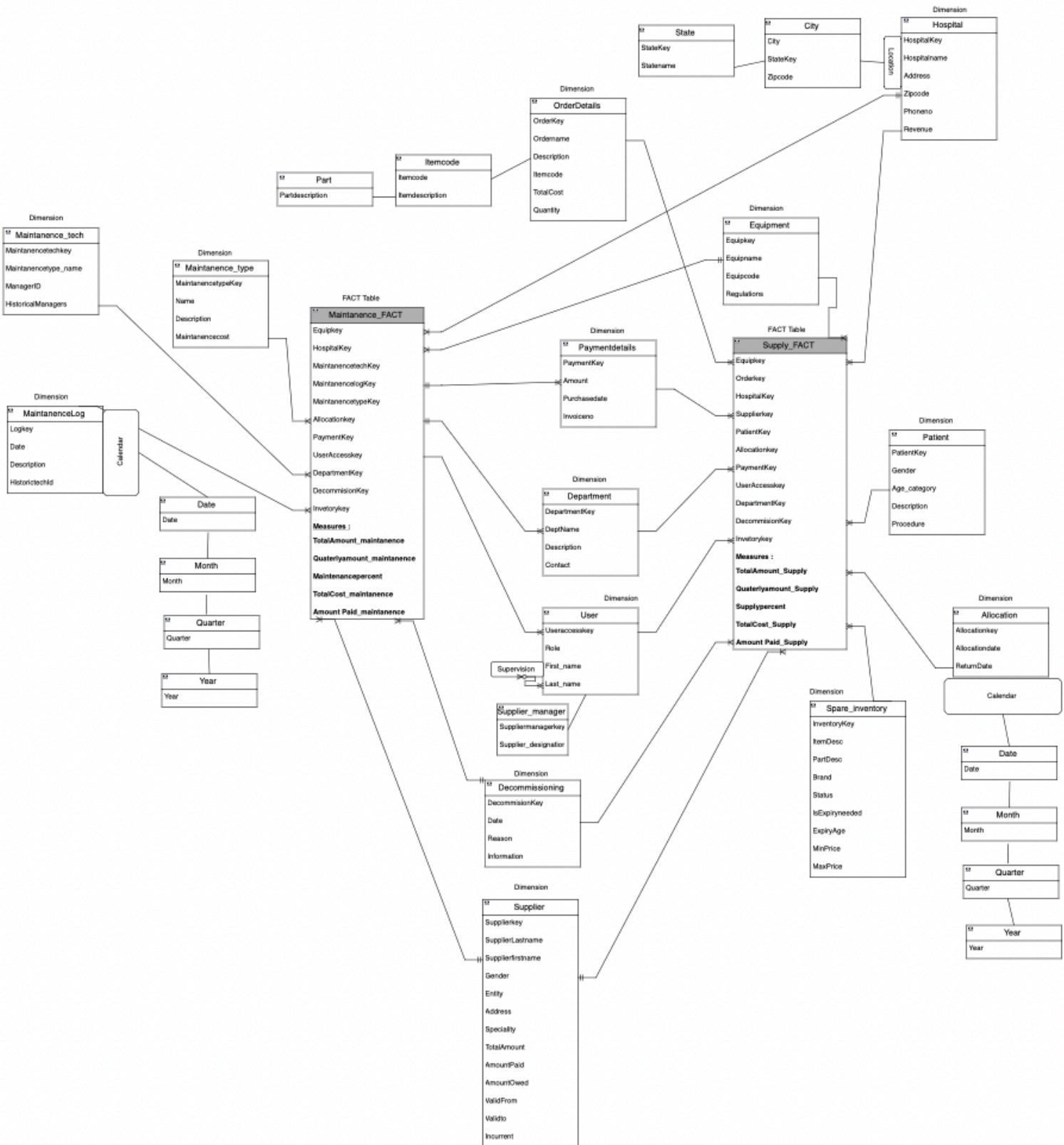
We aim to build a centralized data warehouse with constellation schema as we have two fact tables – **Supply_fact** and **Maintanence_Fact**. This will contain key dimensions like **Supplier dimensions** (Name, address, company, entity), **Maintanence dimensions** like **Maintanence_type**, **maintanence_log**, **maintanence_technician** **Equipment dimensions** (equipmentID, equipment name, productcode, brand) , **User Dimension** which contains everyone who have access to the medical equipments like Owner / Staff -> Supplier part of every healthcare facility/hospital. We further have **orderdetails** and **paymentdetails** as separate dimensions which will keep track of the purchases and the transactions. Next, along with equipment, we also have the inventory of the spare parts required to be constantly replaced or maintain a medical equipment. So, the **inventory dimension** further has item description -> product description.

Facts:

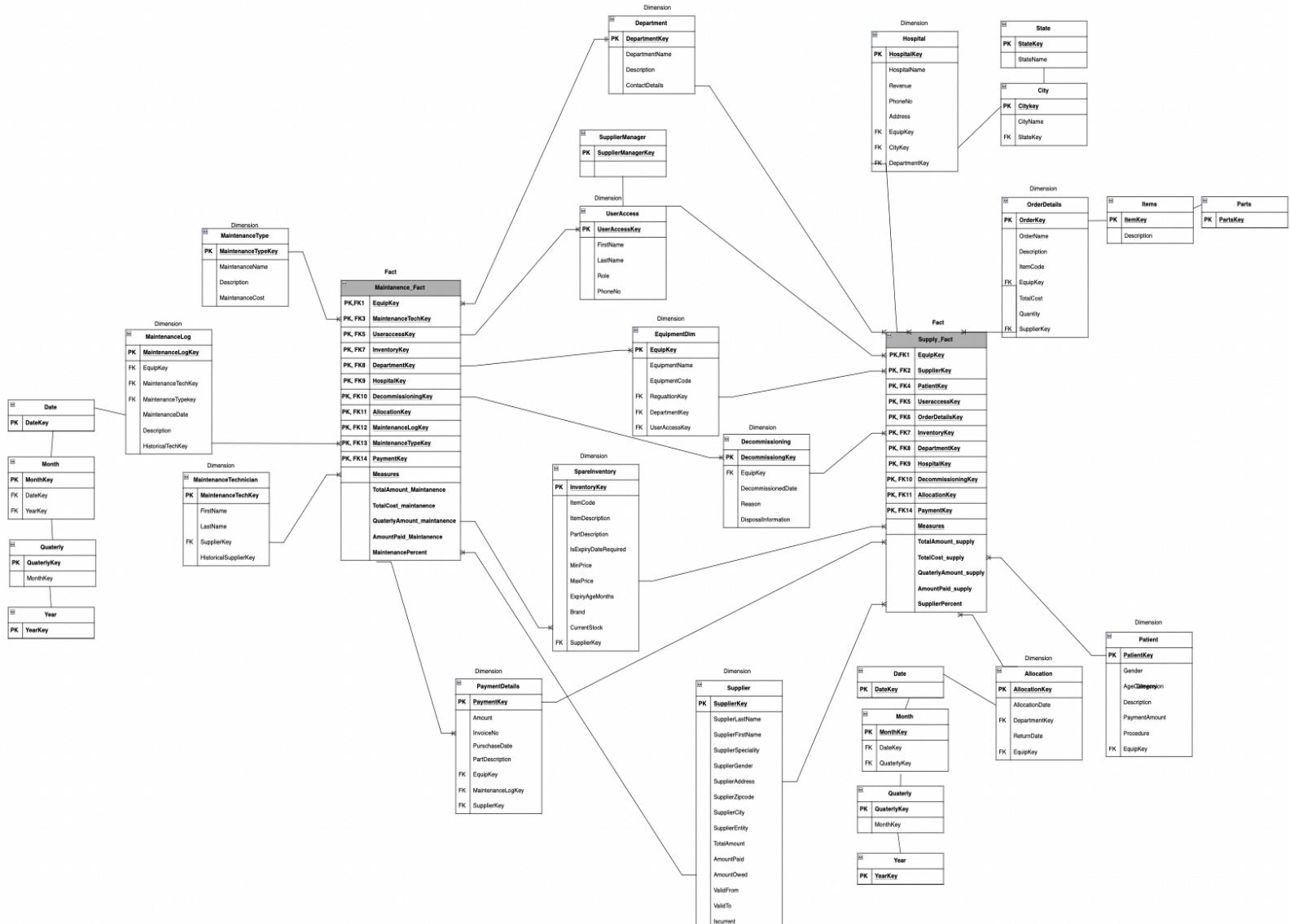
To ensure that we provide optimum maintenance and scheduling, we create a **Maintenance_fact** which have dimensions such as maintanence_log, maintenance_type, maintanance_technician, payment_details, allocation, and include the measures regarding the costs which include the maintenance cost, parts/ supply cost, technician service cost and the maintenance_type cost. This ensures that we have one target table which consolidates all the details regarding the equipments, and the costs expended for each. Secondly, we emphasize on optimum Supply, for which we have **Supply_fact Table** with dimensions such as hospital, supplier, supplier_manager, orderdetails and measures will be revolving around total amount for supply, cost to company from supply, supply revenue percentage.

This multidimensional model allows easy exploration of equipment data for analysis from various perspectives. Another suggestion which we aim to do is to create a alerts and notification to the owners of the equipments to indicate the maintenance or expiry. Each dimension here represents a different aspect of data, and we can analysis critical aspects of healthcare operations.

Multidim Conceptual Model:



Multidim Logical Model:



Schema Implementation in Postgres:

DDL statements implementation of Dimensions and Facts :

```

--1. HospitalDim
CREATE TABLE HospitalDim (
    HospitalKey SERIAL,
    HospitalName VARCHAR(255),
    Address VARCHAR(255),
    CityKey INT,
    EquipmentKey INT,
    DepartmentKey INT,
    State VARCHAR(255),
    Zipcode VARCHAR(255),
    PhoneNo VARCHAR(255),
    Revenue FLOAT
);

--2. SupplierDim
CREATE TABLE medicaldw.SupplierDim (
    SupplierKey SERIAL,
    SupplierLastName VARCHAR(255),
    SupplierFirstName VARCHAR(255),
    SupplierGender VARCHAR(255),
    ContactInformation TEXT,
    SuppliersSpecialty VARCHAR(255),
    Address VARCHAR(255),
    ZipCode VARCHAR(255),
    City VARCHAR(255),
    SupplierEntity VARCHAR(255),
    TotalAmount FLOAT,
    AmountPaid FLOAT,
    AmountOwed FLOAT
);

```

Name	Value
Updated Rows	0
Query	--23. MaintenanceFact CREATE TABLE medicaldw.MaintenanceFact (
	MaintenanceFactKey SERIAL, EquipKey INT, MaintenanceTechKey INT, MaintenanceLogKey INT, UseraccessKey INT, OrderDetailKey INT

```

--2. SupplierDim
CREATE TABLE medicaldw.SupplierDim (
    SupplierKey SERIAL,
    SupplierLastName VARCHAR(255),
    SupplierFirstName VARCHAR(255),
    SupplierGender VARCHAR(255),
    ContactInformation TEXT,
    SuppliersSpecialty VARCHAR(255),
    Address VARCHAR(255),
    ZipCode VARCHAR(255),
    City VARCHAR(255),
    SupplierEntity VARCHAR(255),
    TotalAmount FLOAT,
    AmountPaid FLOAT,
    AmountOwed FLOAT,
    ValidFrom DATE,
    ValidTo DATE,
    IsCurrent BOOLEAN
);

--3. Maintenance Tech Dim
CREATE TABLE medicaldw.MaintenanceTechDim (
    MaintenanceTechKey SERIAL,
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    SupplierKey INT,
    HistoricalSupplierKey INT
);

--4. Department Dim
CREATE TABLE medicaldw.DepartmentDim (
    DepartmentKey SERIAL,
    DepartmentName VARCHAR(255),
    Description TEXT,
    ContactInformation TEXT
);

```

IE7374 Data Warehousing and Data Integration

Dimensions created: (few samples below)

- Allocation Dimension:

The screenshot shows a database interface with a query editor and a results grid. The query editor contains the following SQL code:

```
select * from medicaldw.allocationdim a ;
```

The results grid is titled "allocationdim 1" and displays columns: allocationkey, allocationdate, returndate, equipmentkey. A tooltip indicates: "Enter a SQL expression to filter results (use Ctrl+Space)".

- Decommission Dimension:

The screenshot shows a database interface with a query editor and a results grid. The query editor contains the following SQL code:

```
select * from medicaldw.decommissioningdim d ;
```

The results grid is titled "decommissioningdim 1" and displays columns: decommissionkey, equipkey, decommissioningdate, reason, disposalinformation. A tooltip indicates: "Enter a SQL expression to filter results (use Ctrl+Space)".

- Equipment Dimension:

The screenshot shows a database interface with a query editor and a results grid. The query editor contains the following SQL code:

```
select * from medicaldw.equipmentdim e ;
```

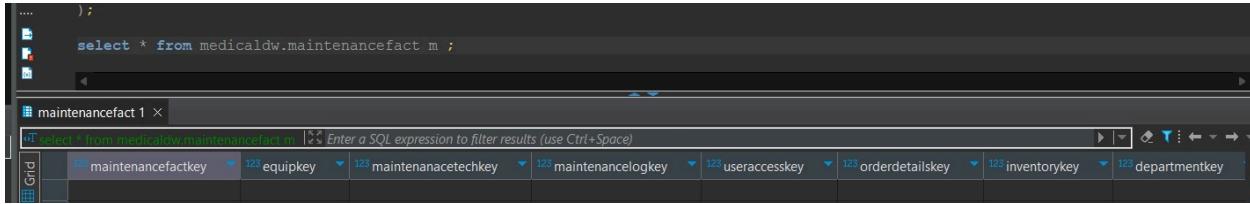
The results grid is titled "equipmentdim 1" and displays columns: equipmentkey, equipmentname, equipmentcode, regulationkey, departmentkey, userkey. A tooltip indicates: "Enter a SQL expression to filter results (use Ctrl+Space)".

- Hospital Dimension:

The screenshot shows a database interface with a query editor and a results grid. The query editor contains the following SQL code:

```
select * from medicaldw.hospitaldim h ;
```

The results grid is titled "hospitaldim 1" and displays columns: hospitalkey, hospitalname, address, citykey, equipkey, departmentkey, state, zipcode, phoneno, revenue. A tooltip indicates: "Enter a SQL expression to filter results (use Ctrl+Space)".

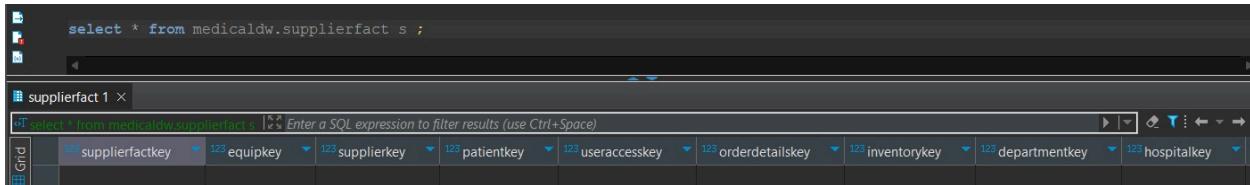
Fact tables:


A screenshot of a database management system interface showing the results of a SQL query. The query is:

```
... );
select * from medicaldw.maintenancefact m;
```

The results are displayed in a table titled "maintenancefact 1". The columns are:

maintenancefactkey	equipkey	maintenanacetechkey	maintenanceologkey	useraccesskey	orderdetailskey	inventorykey	departmentkey
123	123	123	123	123	123	123	123



A screenshot of a database management system interface showing the results of a SQL query. The query is:

```
... );
select * from medicaldw.supplierfact s;
```

The results are displayed in a table titled "supplierfact 1". The columns are:

supplierfactkey	equipkey	supplierkey	patientkey	useraccesskey	orderdetailskey	inventorykey	departmentkey	hospitalkey
123	123	123	123	123	123	123	123	123

SCD implementation:

We have implemented SCD type 2 in Supplier dimension to keep track of the supplier validity and update in case their details change and SCD type 3 in maintenance technician dimension so that we can keep track of the previous and current data. Further Maintenance log also requires frequent updating and analysis; therefore, we have implemented SCD type 3 here. These are implemented to capture the changes in assignments and their IDs.

Primary Events:

For medical equipment supply and maintenance, primary events include various transactions and occurrences related to supply/procurement, maintenance, and usage of medical equipment. Here, equipment acquisition marks the beginning of the equipment's lifecycle, and this is computed usually in the beginning of a quarter. So here we are focusing for the year 2023 and the acquisition from 2023 are in major focus. The supply revenue generated every quarter is always 55% greater than then maintenance revenue. The various types of machines include MRI, ventilators, beds from various vendors who can be individual(doctor) or an organization. Therefore, keeping a track of the quarterly amount is essential. Supplier can be an individual, the doctor or organizations to be a supplier. We have noticed from previous years, that doctor suppliers are more than organizations suppliers. Next, we move towards the equipment installation at a healthcare facility, which consists of details such as date, location, technicians, and this can take up many labor hours, an ICU installation takes around 4 hours, and the technician cost is close to 600\$. Here, the technician's cost is directly proportional to the experience, i.e., a technician with more than 5 years' experience is assigned to a more challenging task thereby charging more. Users, such as medical staff, technicians, patients request for these to be used, for which we have logs which consists of details such as priority, user, date. Usually, the patients and technicians get high priority for treatment usage and maintenance. These logs are consisting of data for the current year (2023) and the others get truncated. Maintaining spare parts inventory efficiently is crucial and involves tracking of availability and consumption. Therefore, for the hospitals, it is crucial to keep track of the supply and maintenance percent of the total cost to hospital, previous observations indicate that supply cost percentage is usually greater by a large margin.

Tracking the frequent usage of equipment can help in scheduling and allocation where we have details on where the equipment is allocated and when it is returned. Here, we need to ensure that the decommissioned equipments aren't being used or allocated that is decommissioned date is 10/10/2023, the equipment should have been returned before. Other than this, communication with the suppliers also is documented which have the regulations, negotiations, and the costs template.

By capturing all these events in warehouse, we can gain valuable insights into equipment supply, maintenance, cost optimization based on various departments/ categories, patient safety and regulatory standards. This data can help make data driven decisions, optimize cost and maintenance schedules and enhance the overall system.

OLAP Operations:

1. What is the quarterly revenue generated by each supplier, and how does it vary over the months?

```
Res1 <- ROLLUP*(Supply_FACT, Supplier-> Name, Supplier -> OrderDate ,      Sum(TotalCost) as  
MonthlyRevenue)
```

2. How does the equipment maintenance cost vary across different types of equipment? (change)

```
Res1 <- DRILLACROSS (Supply_FACT, Maintanence_FACT)  
ROLLUP* (Res1, Equipmentname -> Name, maintanencetype-> maintanencetypename, AVG(maintanencecost))
```

3. Which department is the most cost-effective for supply cost, and which suppliers have the highest outstanding amounts?

```
Res1 <- ROLLUP*( Supply_FACT , Department -> deptname, supplier -> suppliername , sum(amountowed) as  
outstandingamount, sum(totalcost) as totalsupplycost, max(totalcost) by department )
```

4. What is the average age category of patients receiving different procedures, and how does it vary by gender?

```
Res1 <- DRILLDOWN (Supply_FACT, patient -> procedure, patient-> gender)
```

```
Res2 <- ROLLUP* (res1, patient-> Gender, count(agecategory) by patient)
```

5. What is the current spare inventory status, and how does it affect equipment availability?

```
Res1 <- DICE(Supply_FACT , (status = 'Active' AND currentstock != 0 AND returndate <= currentdate))
```

6. How does the maintenance cost vary for a maintenance technician of 5 years and more experience?

```
Res1 <- DICE(Maintanence_FACT, maintanencetech->maintanencetechname, maintanencetype -> cost  
,(experience > 5))
```

```
Res2 <- ROLLUP* (Res1, maintanencetechname -> name, Maintanencetypename -> experience,  
sum(maintanencecost))
```

7. List the user access levels and the types of access they have to the equipment.

Res1 <- DRILLDOWN(Supply_FACT, useraccess-> user, supplier ->supplier)

Res2 <- ROLLUP*(Res1, Useraccess-> role, supplier->entity, count(role) by entity, role)

8. What percentage of maintenance amount constitute the total payment amount given?

Res1 <- DRILLDOWN (Maintanence_FACT ,paymentdetails -> totalamount, supplier -> amountgiven,(
((Totalamount – amountgiven)*100)/Totalamount))

9. Is there any equipment that has been allocated after being decommissioned?

Res1 <- DICE (Supply_FACT, decommission -> date, allocation -> allocationdate, allocation -> returndate, (allocationdate < decommissiondate AND returndate < decommisiondate))

Res2 <- ROLLUP * (Res1, equipment -> name, COUNT(equipmentid))

Warehouse Design Proposal:

Problem Definition:

Our main aim is to implement a comprehensive data solution by creating a centralized repository for all equipment related data. This will help in ensuring proactive maintenance, cost control, inventory optimization and regulate compliance. Here, we further aim to create a scheduling table which will regularize maintenance schedules for each equipment and follow the regulation. Furthermore, category/department wise segregation of equipment will ease the process of tracking and ensure the entire system is reliable.

Dimensions:

We aim to build a centralized data warehouse with constellation schema as we have two fact tables – **Supply_fact** and **Maintanence_Fact**. This will contain key dimensions like **Supplier dimensions** (Name, address, company, entity), **Maintanence dimensions** like **Maintanence _type, maintanence_log, maintanence_technician**, **Equipment dimensions** (equipmentID, equipment name, productcode, brand), **User Dimension** which contains everyone who have access to the medical equipments like Owner / Staff -> Supplier part of every healthcare facility/hospital. We further have **orderdetails and paymentdetails** as separate dimensions which will keep track of the purchases and the transactions. Next, along with equipment, we also have the inventory of the spare parts required to be constantly replaced or maintain a medical equipment. So, the **inventory dimension** further has item description -> product description.

Facts:

To ensure that we provide optimum maintenance and scheduling, we create a **Maintenance_fact** which have dimensions such as **maintanence_log, maintenance_type, maintanance_technician, payment_details, allocation**, and include the measures regarding the costs which include the maintenance cost, parts/ supply cost, technician service cost and the maintenance type cost. This ensures that we have one target table which consolidates all the details regarding the equipments, and the costs expended for each. Secondly, we emphasize on optimum Supply, for which we have **Supply_fact Table** with dimensions such as hospital, supplier, supplier_manager, orderdetails and measures will be revolving around total amount for supply, cost to company from supply, supply revenue percentage.

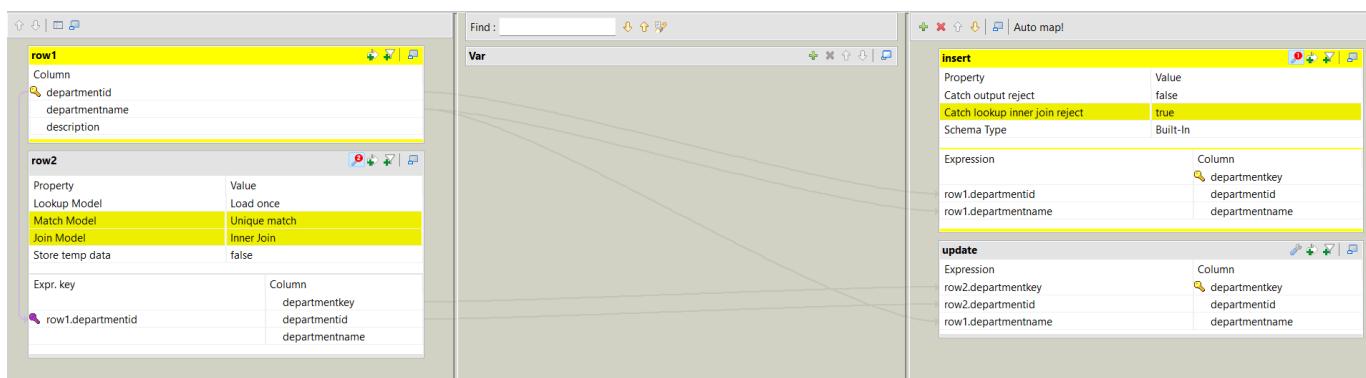
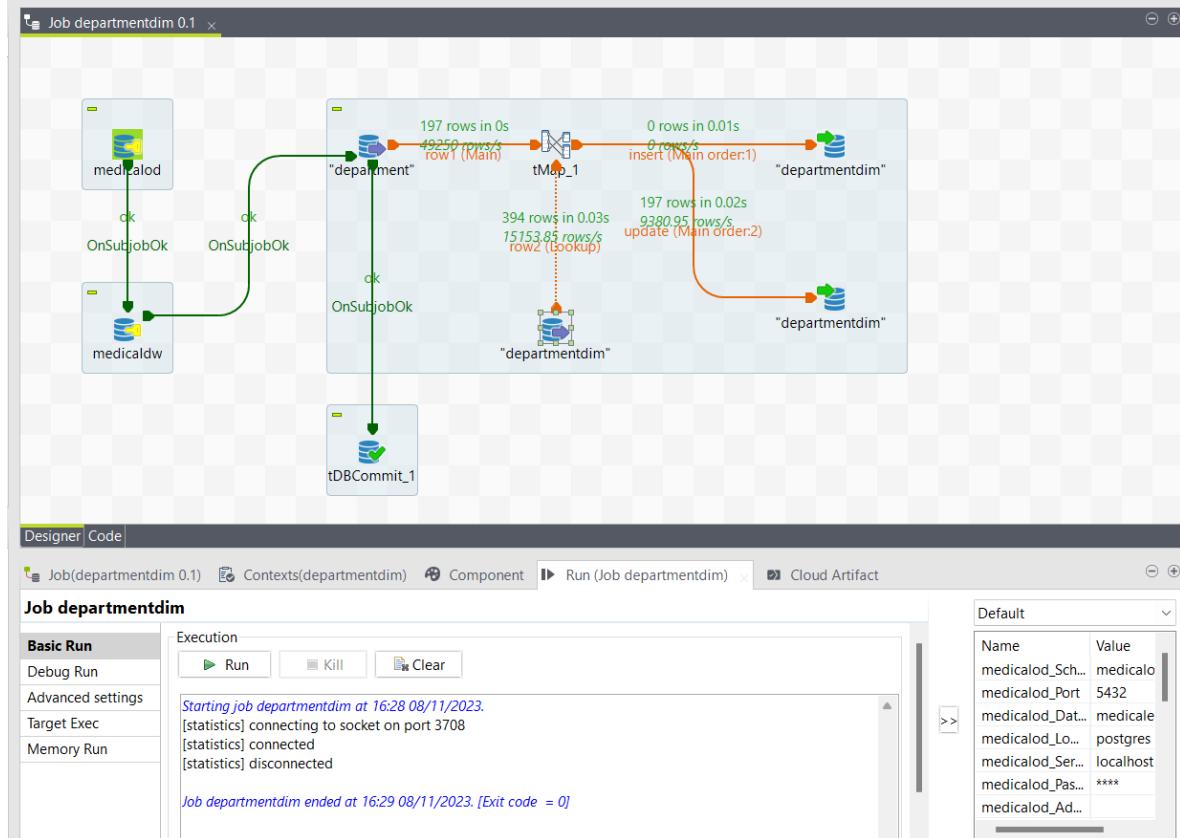
SCD:

We have implemented SCDs in supplier – Type 2, Maintenance tech – Type 3 and Maintenance log – Type 3. In supplier, we have implemented Type 2 SCD to keep track of the amount paid and owed to the suppliers. In maintenance tech, we use type 3 SCD to keep track of the supplier ID changes. In maintenance log, we use type 3 SCD to update and keep a track of the maintenance type ID.

We have further analytical queries to ensure optimum analysis and insights.

Data and Control flows implemented in Talend:

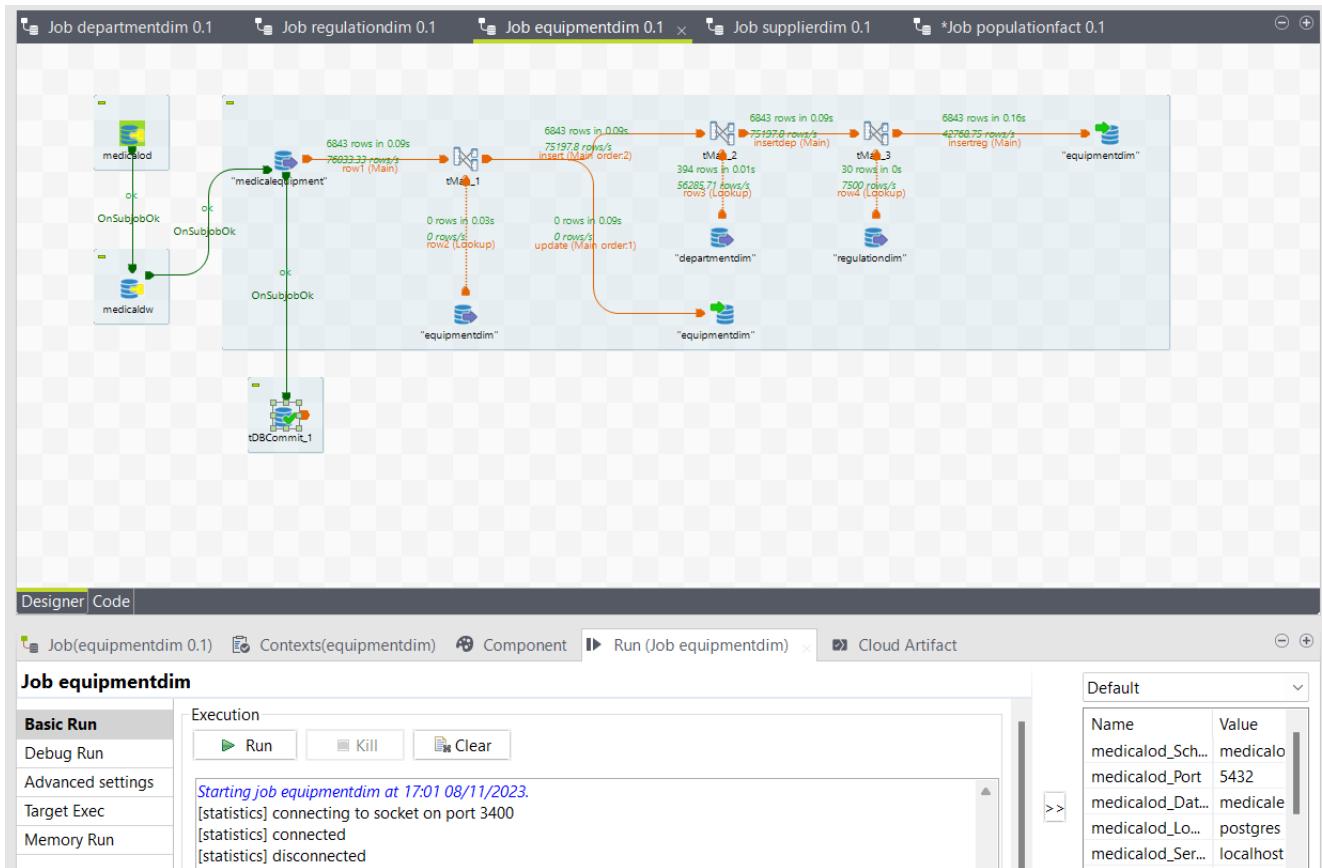
1. Department Dim



departmentdim | Enter a SQL expression to filter results (use Ctrl+Space)

	123 departmentkey	123 departmentid	ABC departmentname
1	1	1	Acupuncturist
2		2	Addiction Medicine
3		3	All Other Suppliers
4		4	Allergy/Immunology
5		5	Ambulatory Surgical Center
6		6	Anesthesiologist Assistants
7		7	Anesthesiology
8		8	Assisted Living Facility
9		9	Audiologist (billing independently)
10		10	Audiologist-Hearing Aid Fitter
11		11	Behavioral Analyst
12		12	Cardiac Electrophysiology
13		13	Cardiac Surgery
14		14	Cardiology
15		15	Case Management

2. Equipment Dim:



IE7374 Data Warehousing and Data Integration

The screenshot shows the Talend Data Integration interface. On the left, the 'Mapping Designer' panel displays two rows of data:

- row1:** Columns equipmentid, equipmentname, equipmentcode, regulationid, departmentid, userid.
- row2:** Properties and Values. Lookup Model: Load once; Match Model: Unique match; Join Model: Inner Join; Store temp data: false. Expr. key: row1.equipmentid; Column: equipmentkey, equipmentid, equipmentname, equipmentcode, regulationkey, departmentkey.

The central area is the 'Previewer' showing the mapping logic with various nodes and connections. On the right, the 'Auto map!' panel shows update and insert statements:

```

update
Expression          Column
row2.equipmentkey   equipmentkey
row2.equipmentid    equipmentid
row2.equipmentname  equipmentname
row2.equipmentcode  equipmentcode
row2.regulationkey  regulationkey
row2.departmentkey   departmentkey

insert
Property          Value
Catch output reject false
Catch lookup inner join reject true
Schema Type        Built-In
Expression          Column
row1.equipmentid   equipmentkey
row1.equipmentname  equipmentid
row1.equipmentcode  equipmentname
row1.regulationid  equipmentcode
row1.departmentid   departmentkey

```

	123 equipmentkey	123 equipmentid	ABC equipmentname	ABC equipmentcode	123 regulationkey	123 departmentkey
1	1	1	Protector, Dental	BRW	1	1
2	2	2	Stool, Anesthesia	BRX	2	2
3	3	3	Cabinet, Table And Tray, Anesthesia	BRY	3	3
4	4	4	Analyzer, Gas, Helium, Gaseous-Phase	BSE	4	4
5	5	5	Absorber, Carbon-Dioxide	BSF	5	5
6	6	6	Algesimeter, Powered	BSI	6	6
7	7	7	Mask, Gas, Anesthetic	BSJ	7	7
8	8	8	Cuff, Tracheal Tube, Inflatable	BSK	8	8
9	9	9	Filter, Conduction, Anesthetic	BSN	9	9
10	10	10	Catheter, Conduction, Anesthetic	BSO	10	10
11	11	11	Needle, Conduction, Anesthetic (W/Wo Introducer)	BSP	11	11
12	12	12	Stylet, Tracheal Tube	BSR	12	12
13	13	13	Catheters, Suction, Tracheobronchial	BSY	13	13
14	14	14	Gas-Machine, Anesthesia	BSZ	14	14
15	15	15	Hook, Ether	BTB	15	15

3. Supplier Dim:

The screenshot shows the Talend Data Integration interface. At the top, the job tabs are: Job departmentdim 0.1, Job regulationdim 0.1, Job equipmentdim 0.1, Job supplierdim 0.1 (highlighted), and *Job populationfact 0.1.

The main area shows a job flow diagram for 'Job supplierdim 0.1'. It starts with a 'medicalod' source, followed by a 'tMap_1' component (with statistics: 14998 rows in 2.21s, 6799.73 rows/s, row1 (Main)). This is followed by a 'datainsert' component (with statistics: 14998 rows in 2.26s, 6636.29 rows/s, datainsert (Main)) and a 'supplierdim' target. The flow ends with a 'tDBCommit_1' component. Subjob connections are labeled 'OnSubjobOk'.

At the bottom, the 'Job supplierdim' tab is active, showing the execution history:

- Execution: Run, Kill, Clear.
- Log: Starting job supplierdim at 18:59 08/11/2023. [statistics] connecting to socket on port 4076 [statistics] connected [statistics] disconnected
- Message: Job supplierdim ended at 18:59 08/11/2023. [Exit code = 0]

On the right, the 'Cloud Artifact' panel shows environment variables:

Name	Value
medicalod_Sch...	medicalod
medicalod_Port	5432
medicalod_Dat...	medicale
medicalod_Lo...	postgres
medicalod_Ser...	localhost
medicalod_Pas...	****
medicalod_Ad...	

Tmap:

row1

Column
supplierid
supplierlastname
supplierfirstname
supplierdesignation
suppliergender
suppliernet
supplieraddress
suppliercity
supplierstate
supplierzipcode
supplierspecialty
totalamount
amountpaid
amountowed

Var

Expression	Type	Variable
row1.supplierfirstname + " " + row1.sup...	String	suppliername

datainsert

Column
supplierid
suppliername
supplierdesignation
suppliergender
suppliernet
supplieraddress
suppliercity
supplierstate
supplierzipcode
supplierspecialty
totalamount
amountpaid
amountowed

Type-2 SCD:

SCD component editor

Unused

- supplierdesignation
- suppliernet
- suppliergender
- suppliername

Type 0 fields

- supplieraddress
- suppliercity
- supplierstate
- supplierzipcode

Type 1 fields

- amountowed
- amountpaid
- totalamount

Type 2 fields

type	name	creation	compleme...
start	validfrom	Job start time	
end	validto	NULL	
<input type="checkbox"/> versi...	scd_version		
<input checked="" type="checkbox"/> active	iscurrent		

Versioning

Type 3 fields

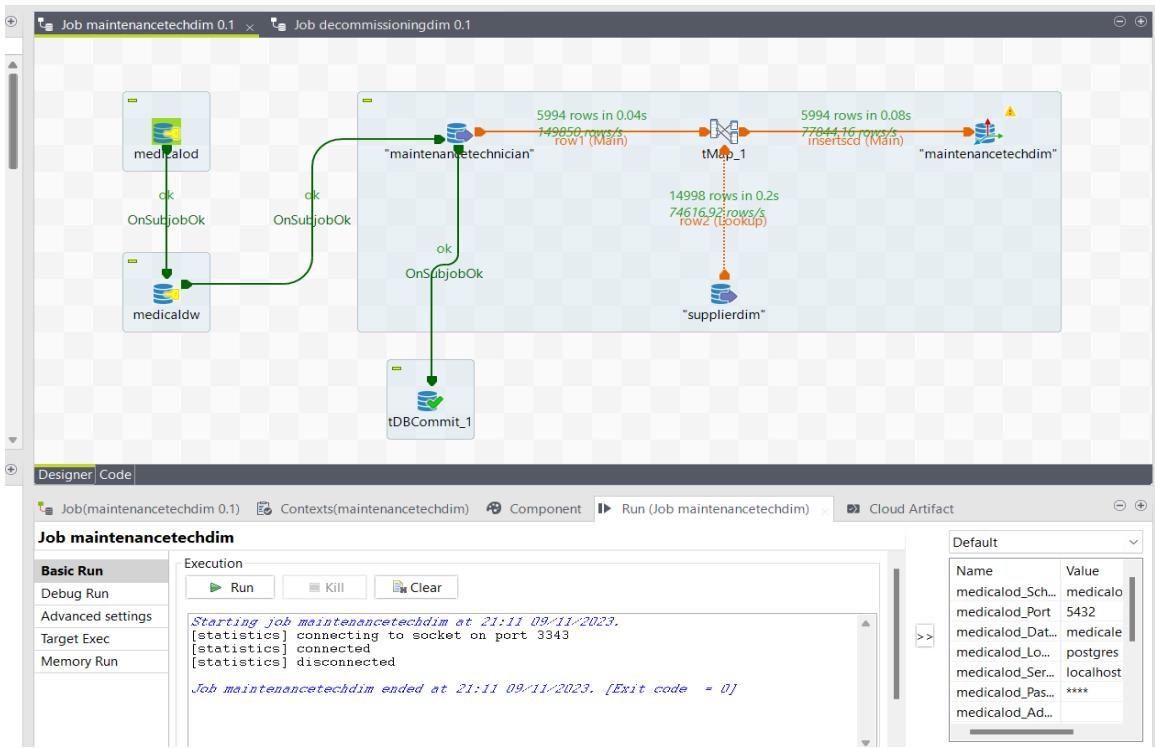
current value	previous value
---------------	----------------

OK **Cancel**

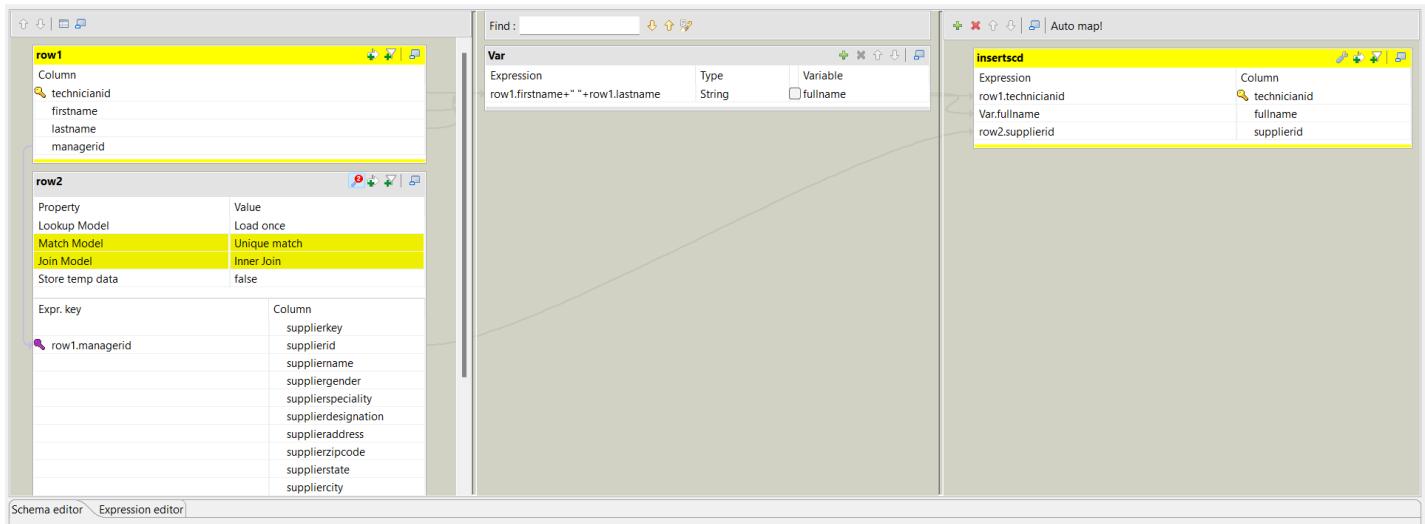
IE7374 Data Warehousing and Data Integration

	123_supplierkey	123_supplierid	abc_suppliername	abc_suppliergender	abc_supplierspecialty	abc_supplierdesignation	abc_supplieraddress
1	1	1	ARDALAN ENKESHAJI	M	Internal Medicine	M.D.	900 SETON DR
2	2	2	RASHID KHALIL	M	Anesthesiology	M.D.	4126 N HOLLAND SYLVANIA RD
3	3	3	DAVID GIRARDI	M	Family Practice	D.O.	456 MAGEE AVE
4	4	4	FREDERICK WEIGAND	M	Family Practice	MD	1565 SAXON BLVD STE 202
5	5	5	AMANDA SEMONCHE	F	Internal Medicine	DO	1021 PARK AVE
6	6	6	OTNIEL HERNANDEZ	M	Nurse Practitioner	DNP, FNP	4410 W 16TH AVE
7	7	7	JAIVANTI LOHANO	F	Family Practice	MD	2215 PORTLAND AVE
8	8	8	MARK STELLINGWORTH	M	Cardiology	MD	8 RICHLAND MEDICAL PARK DR
9	9	9	MICHAEL VIGGIANELLI	M	Family Practice	MD	HWY 1 CALIFORNIA MENS COLO
10	10	10	CHARLES STEVENS	M	Anesthesiology	M.D.	1665 S IMPERIAL AVE STE D
11	11	11	DAVID KANTER	M	Physical Medicine and Rehabilitation	MD	750 E ADAMS ST
12	12	12	JACLYN JONES	F	Orthopedic Surgery	D.O.	2250 DREW ST
13	13	13	LUCILLE SAHA	F	Family Practice	M.D.	1397 S LINDEN RD
14	14	14	LAKSHMI SRINIVASAN	F	Endocrinology	MD	3200 KEARNEY ST
15	15	15	PATRICIA HOPKINS	F	Internal Medicine	M.D.	500 CONGRESS ST STE 1B

4. MaintenanceTechDim:



Tmap:



	123 maintenancetechkey	123 technicianid	ABC fullname	123 supplierid	123 historicalsupplierid
1		1	Calvin Johnson	1	[NULL]
2		2	Harry Cleveland	2	[NULL]
3		3	John Nixon	3	[NULL]
4		4	Franklin Jefferson	4	[NULL]
5		5	Harry Eisenhower	5	[NULL]
6		6	Grover Garfield	6	[NULL]
7		7	Benjamin Grant	7	[NULL]
8		8	Harry Hayes	8	[NULL]
9		9	Ulysses Buchanan	9	[NULL]
10		11	Richard Johnson	11	[NULL]

Type-3 SCD:

SCD component editor

The SCD component editor interface displays the following fields categorized by type:

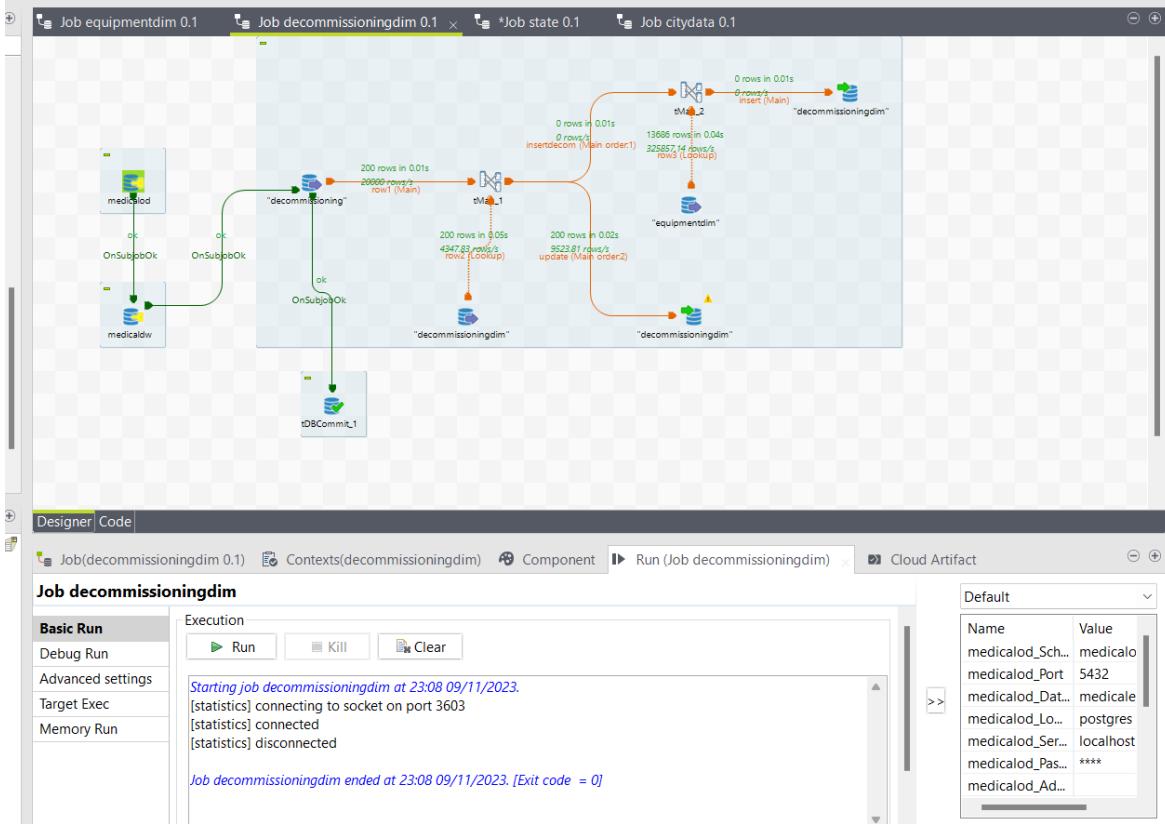
- Type 0 fields:** fullname
- Type 1 fields:** (empty)
- Type 2 fields:** (empty)
- Versioning:**

type	name	creation	comple...
start	scd_start	Job start time	
end	scd_end	NULL	
<input type="checkbox"/> versi...	scd_version		
<input type="checkbox"/> active	scd_active		
- Type 3 fields:**

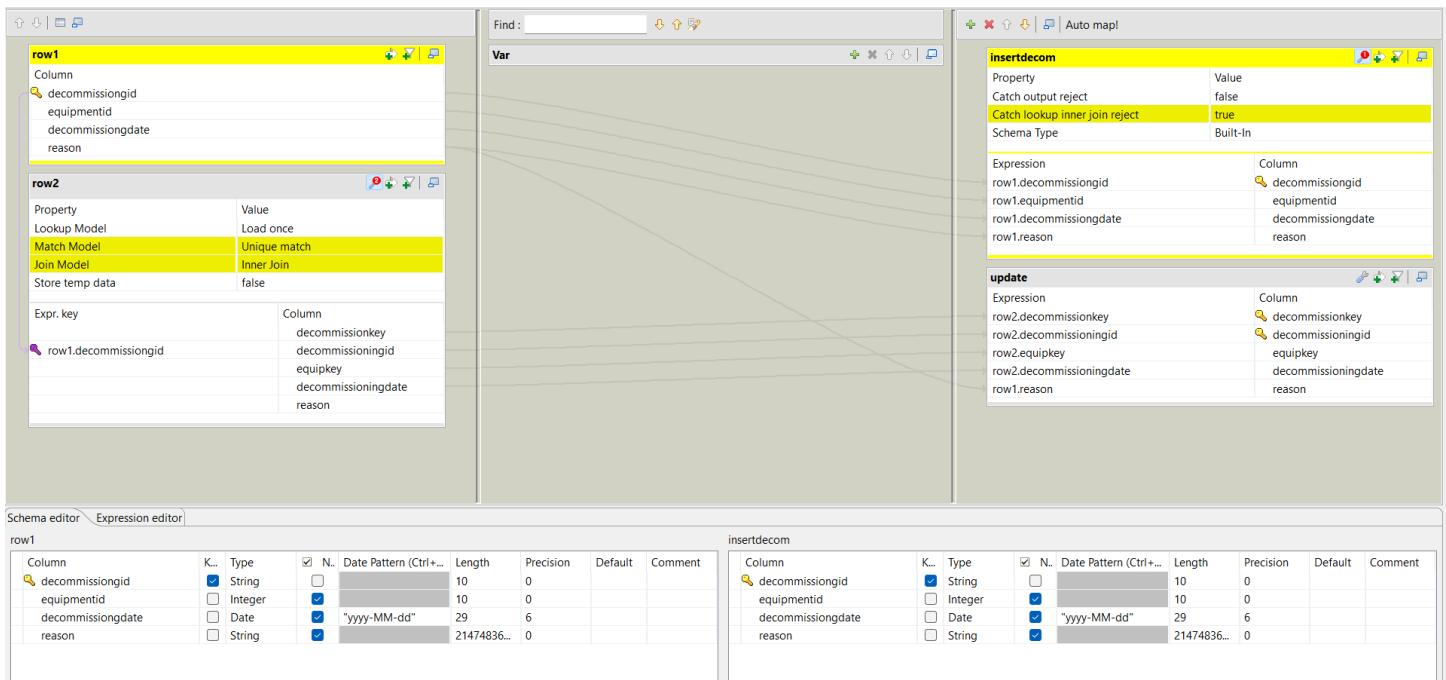
current value	previous value
supplierid	historicalsupplierid

Buttons at the bottom right: OK and Cancel.

5. DecommissioningDim:



Tmap-1:



Tmap-2:

The screenshot shows the Talend Data Integration interface with three main panels:

- Left Panel:** Displays the "insertdecom" component configuration. It includes a "Column" section with columns: decommissioningid, equipmentid, decommissioningdate, and reason. Below it is a "row3" section with properties: Lookup Model (Load once), Match Model (Unique match), Join Model (Inner join), and Expr. key (insertdecom.equipmentid). A detailed view of the Expr. key shows columns: equipmentkey, equipmentid, equipmentname, equipmentcode, regulationkey, and departmentkey.
- Middle Panel:** Shows the mapping between "insertdecom" and "insert". The "insert" component has columns: decommissioningid, equipkey, decommissioningdate, reason, and decommissionkey. Arrows indicate the flow from "insertdecom" to "insert".
- Bottom Panel:** Displays the "Schema editor \ Expression editor" for both "insertdecom" and "insert" components. Both schemas have columns: decommissioningid (String, 10, 0), equipmentid (Integer, 10, 0), decommissioningdate (Date, "yyyy-MM-dd", 29, 6), reason (String, 21474836, 0), and decommissionkey (Integer, 10, 0).

Data Preview:

	decommissionkey	decommissioningid	equipkey	decommissioningdate	reason
1		DC001	3,229	2023-11-14	Upgrade
2		DC002	3,627	2023-08-31	Upgrade
3		DC003	3,560	2023-11-10	Upgrade
4		DC004	5,449	2023-07-14	Safety concerns
5		DC005	2,238	2023-07-15	Equipment malfunction
6		DC006	2,851	2023-08-22	End of life
7		DC007	3,601	2023-11-25	End of life
8		DC008	1,923	2023-02-16	Upgrade
9		DC009	4,352	2023-03-14	End of life
10		DC010	999	2023-08-19	End of life
11		DC011	124	2023-03-13	Equipment malfunction
12		DC012	5,253	2023-09-24	Obsolete technology
13		DC013	1,902	2023-12-01	Equipment malfunction
14		DC014	1,893	2023-10-18	Equipment malfunction
15		DC015	3,562	2023-10-30	Equipment malfunction

6. State:

The screenshot shows a Talend job configuration with the following details:

- Job Components:**
 - medicalod (Subjob): Contains medicallod and medicaldw components.
 - state (Main Job): Contains state data, tMap_1, and state database components.
 - tDBCommit_1 (Commit): Commits changes to the database.
- Flows:**
 - medicalod feeds into state via state data.
 - state data feeds into tMap_1.
 - tMap_1 feeds into state database.
 - tDBCommit_1 follows the state database component.
- Metrics:**
 - 55 rows in 0s (row1 (Main))
 - 13750 rows/s (rate)
 - 55 rows in 0.01s (row1 (Main))
 - 61141 rows/s (rate)
 - 0 rows in 0.02s (row2 (Lookup/Update (Main order:2)))
 - 0 rows/s (rate)
 - 0 rows in 0.01s (row2 (Lookup/Update (Main order:2)))
 - 0 rows/s (rate)
- Job State:**
 - Execution: Run, Kill, Clear.
 - Log: Starting job state at 23:39 09/11/2023. [statistics] connecting to socket on port 3697. [statistics] connected. [statistics] disconnected. Job state ended at 23:39 09/11/2023. [Exit code = 0]
 - Cloud Artifact: Job(state 0.1), Contexts(state), Component, Run (Job state), Cloud Artifact.
 - Default Contexts (Table): Lists variables like medicalod_Sch..., medicalod_Port, medicalod_Dat..., medicalod_Lo..., medicalod_Ser..., medicalod_Pas..., and medicalod_Ad...

Tmap:

The screenshot shows the Talend Data Integration interface with three main panels:

- Left Panel:** Displays the configuration for a Tmap component named "row1". It includes sections for "Column" (statename), "Property" (Lookup Model: Load once, Match Model: Unique match, Join Model: Inner Join, Store temp data: false), and "Expr. key" (row1.statename).
- Middle Panel:** Shows the mapping logic between "row1" and "row2". A "Var" table is also visible.
- Right Panel:** Displays two sections: "insert" and "update". The "insert" section maps "row1.statename" to "statekey" and "statename". The "update" section maps "row2.statekey" and "row1.statename" to "statekey" and "statename".

Preview Tables:

- Table 1 (Top):** statekey | statename

	statekey	statename
1	1	AE
2	2	AK
3	3	AL
4	4	AR
5	5	AZ
- Table 2 (Bottom):** allocationkey | allocationid | allocationdate | returndate | equipmentkey

	allocationkey	allocationid	allocationdate	returndate	equipmentkey
1	1	AL0001	2023-07-25	2023-08-08	1
2	2	AL0002	2023-03-28	2023-04-08	2
3	3	AL0003	2023-09-22	2023-10-07	3
4	4	AL0004	2023-03-15	2023-03-30	4
5	5	AL0005	2023-07-27	2023-08-09	5

7. City:

The screenshot shows the Talend Data Integration Designer interface with the following components and flow:

- Jobs:** Job equipmentdim 0.1, Job decommissioningdim 0.1, Job state 0.1, Job citydata 0.1
- Job Flow:**
 - Starts with a Subjob (medicallod) connecting to a Main job (citydata).
 - The Main job (citydata) has two parallel paths:
 - Path 1: citydata -> tMap_1 -> "city" (target table).
 - Path 2: citydata -> tMap_2 -> "state" (target table).
 - Both "city" and "state" tables feed into a final step (tDBCommit_1).
- Metrics:** Performance metrics are displayed along the data flow, such as "4428 rows in 0.03s" for the first tMap and "0 rows in 0.02s" for the second tMap.
- Job Details:** The "Job citydata" details panel shows the execution status: "Starting job citydata at 23:56 09/11/2023." and "Job citydata ended at 23:56 09/11/2023. [Exit code = 0]".
- Configuration:** The "Default" configuration table lists parameters like medicallod_Sch..., medicallod_Port, medicallod_Dat..., medicallod_Lo..., medicallod_Ser..., medicallod_Pas..., and medicallod_Ad... with their respective values.

Tmap-1:

The screenshot shows the Talend Data Integration environment. At the top, there are three tabs: 'Schema editor', 'Expression editor', and 'tMap'. The 'tMap' tab is active, displaying a complex mapping configuration. On the left, the 'row1' schema is defined with columns 'cityname' and 'statename'. Below it, 'row2' properties are set: 'Lookup Model' to 'Load once', 'Match Model' to 'Unique match', and 'Join Model' to 'Inner Join'. The 'Expr. key' section maps 'row1.cityname' to 'citykey', 'cityname', and 'statekey'. To the right, the 'Auto map!' panel shows two entries: 'cityinsert' and 'update'. 'cityinsert' has a 'Property' column with 'Value' 'false' and 'Catch lookup inner join reject' checked. It also lists 'row1.cityname' and 'row1.statename' under 'Expression'. 'update' has a similar structure. Below these panels is a 'Schema editor' table comparing 'row1' and 'cityinsert' schemas. At the bottom, a preview table shows data with columns 'citykey', 'cityname', and 'statekey'.

Column	Type	N.	Date Pattern (Ctrl+D)	Length	Precision	Default	Comment
cityname	String	✓		14	0		
statename	String	✓		2	0		

Column	Type	N.	Date Pattern (Ctrl+D)	Length	Precision	Default	Comment
cityname	String	✓		14	0		
statename	String	✓		2	0		

	123 citykey	ABC cityname	123 statekey
1	5	NEWARK	2
2	6	DENVER	2
3	7	HASTINGS	2
4	8	LONG BEACH	2
5	12	JOHNSON CITY	2

8. MaintenanceFact :

The screenshot shows a Talend job named 'Job maintenancefact 0.1'. The job flow starts with a 'medicalload' component, followed by a 'medicaldw' component, and ends with a 'tDBCommit_1' component. A green line connects the 'medicalload' and 'medicaldw' components. From 'medicaldw', a green line leads to a 'tMap' component labeled 'tMap_1'. The 'tMap' component is connected to several database components: 'maintenancelogdim', 'maintenancetechnim', 'hospitaldim', 'maintenancetypedim', 'orderdetailsdim', and 'paymentdetailsdim'. Each of these database components has associated statistics displayed above them. For example, 'maintenancelogdim' shows 14000 rows in 0.32s, and 'orderdetailsdim' shows 6000 rows in 0.07s. The 'tMap' component also has its own statistics: 35005.63 rows/s and an 'Insert (Main)' operation. Finally, a red line connects 'tMap_1' to a 'maintenancefact' component, which is shown with a green checkmark indicating success. The bottom of the screen shows the 'Designer' tab is selected, along with other tabs like 'Code', 'Job(maintenancefact 0.1)', 'Contexts(maintenancefact)', 'Component', 'Run (Job maintenancefact)', and 'Cloud Artifact'. The 'Job maintenancefact' window shows a 'Basic Run' section with 'Run', 'Kill', and 'Clear' buttons, and a log area with the message 'Starting job maintenancefact at 16:50 15/11/2023'.

IE7374 Data Warehousing and Data Integration

The screenshot shows the Informatica PowerCenter interface with the Mapping Editor open. On the left, the Schema editor displays two rows of columns:

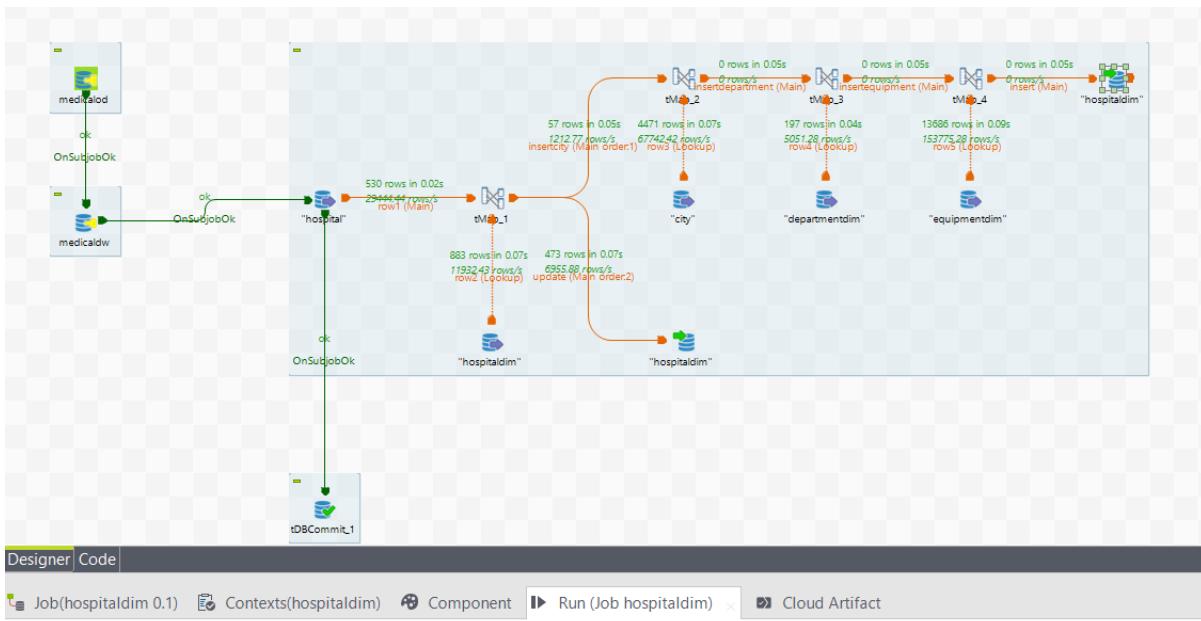
- row1:** logkey, logid, equipkey, maintenancetechid, maintenancedate, maintenancetypeid, description, historicalmaintenancetypeid.
- row2:** Property Value, Lookup Model Load once, Match Model Unique match, Join Model Inner Join, Store temp data false. Expr. key Column: row1.maintenancetechid, Column: maintenancetechkey, Column: technicianid, Column: fullname, Column: supplierid, Column: historicalsupplierid.

The Mapping Editor on the right shows various nodes (e.g., Insert, Join, Filter) connected by arrows, representing the data flow between the source and target tables. A specific node in the mapping is highlighted with yellow, and its properties are shown in the 'Var' (Variable) and 'insert' tabs.

Column	Type	N.	Date Pattern (Ctrl+...)	Length	Precision	Default	Comment
logkey	int			10	0		
logid	Integer			10	0		
equipkey	Integer			10	0		
maintenancetechid	Integer			10	0		
maintenancedate	Date		"dd-MM-yyyy"	13	0		
maintenancetypeid	Integer			10	0		

Column	Type	N.	Date Pattern (Ctrl+...)	Length	Precision	Default	Comment
maintenancefactkey	int			10	0		
maintenancetechkey	Integer			10	0		
maintenancelogkey	Integer			10	0		
orderdetailskey	Integer			10	0		
hospitalkey	Integer			10	0		
paymentkey	Integer			10	0		

9. HospitalDim:



The screenshot shows the Informatica PowerCenter Job history interface for the 'hospitaldim' job. The 'Basic Run' tab is selected, showing the execution details:

- Execution:** Run, Kill, Clear.
- Log Output:**

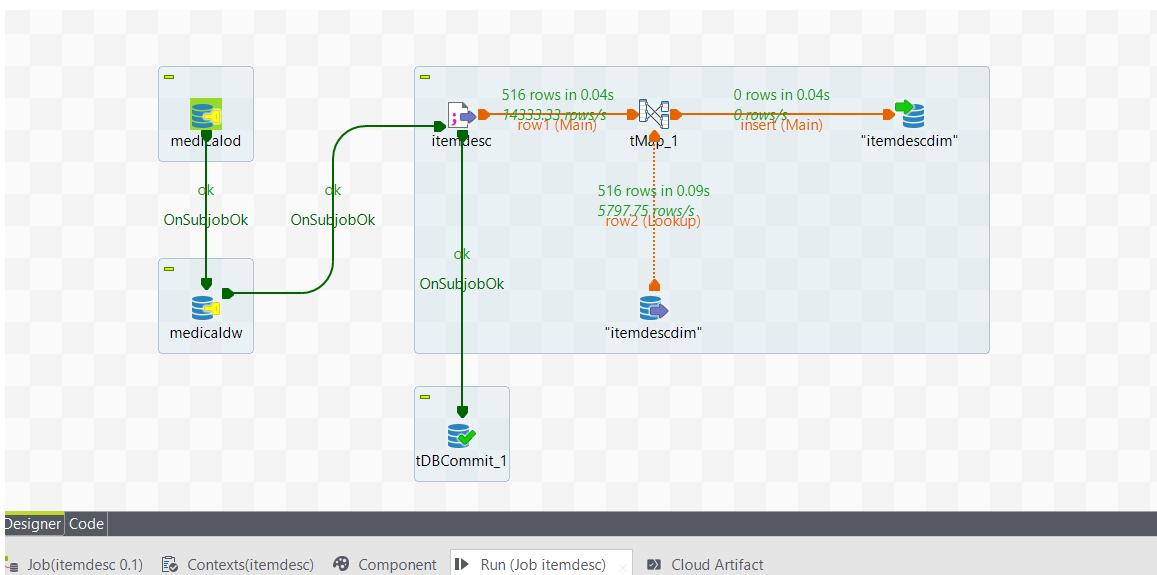
```
Starting job hospitaldim at 18:41 15/11/2023.
[statistics] connecting to socket on port 3609
[statistics] connected
[statistics] disconnected
```
- Job Status:** Job hospitaldim ended at 18:41 15/11/2023. [Exit code = 0]

T-map1 :

The screenshot shows the Talend Data Integration environment with the following components:

- Row1 (Top Left):** A table showing columns for hospitalid, hospitalname, address, city, state, zipcode, phoneno, revenue, equipmentid, and departmentid.
- Var (Top Center):** A variable editor showing properties for the lookup model (Load once), match model (Unique match), and join model (Inner Join). It also shows an expression for Expr.key: row1.hospitalid and columns for hospitalkey, hospitalid, hospitalname, and address.
- Insertcity (Top Right):** A table for inserting data into a city dimension table. It includes columns for hospitalid, hospitalname, address, city, state, zipcode, phoneno, revenue, equipmentid, and departmentid. The "Catch lookup inner join reject" property is set to true.
- update (Bottom Right):** A table for updating the city dimension table based on hospitalkey.
- Schema Editor (Bottom Left):** A table showing the schema for the source table (ow1) with columns hospitalkey, hospitalid, hospitalname, address, zipcode, city, state, and phoneno.
- Insertcity Schema (Bottom Center):** A table showing the schema for the target table (insertcity) with columns hospitalid, hospitalname, address, and zipcode.
- Preview (Bottom):** A grid showing the joined data from Row1 and Var. The columns are hospitalkey, hospitalid, hospitalname, address, zipcode, citykey, equipkey, and depart. The data includes various hospital entries like HEALTHALLIANCE HOSPITALS INC, MOUNT AUBURN HOSPITAL, and STURDY MEMORIAL HOSPITAL.

10. Itemdesc:



The screenshot shows the execution log for the Job itemdesc:

- Execution:** Buttons for Run, Kill, and Clear.
- Log Output:**

```
Starting job itemdesc at 18:43 15/11/2023.
[statistics] connecting to socket on port 3757
[statistics] connected
[statistics] disconnected
Job itemdesc ended at 18:43 15/11/2023. [Exit code = 0]
```

Row Editor (row1) and Row Properties (row2) panels showing configuration for a lookup model:

Column	itemdescription
partdescription	

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Inner Join
Store temp data	false

Expr. key: row1.itemdescription

Column mapping:

itemkey	itemdescription
itemdescription	partdescription

Var panel showing insert configuration:

Property	Value
Catch output reject	false
Catch lookup inner join reject	true
Schema Type	Built-In

insert panel showing expression mapping:

Expression	Column
row1.itemdescription	itemkey
row1.partdescription	partdescription

Schema editor (row1) and Expression editor (row1) showing column definitions:

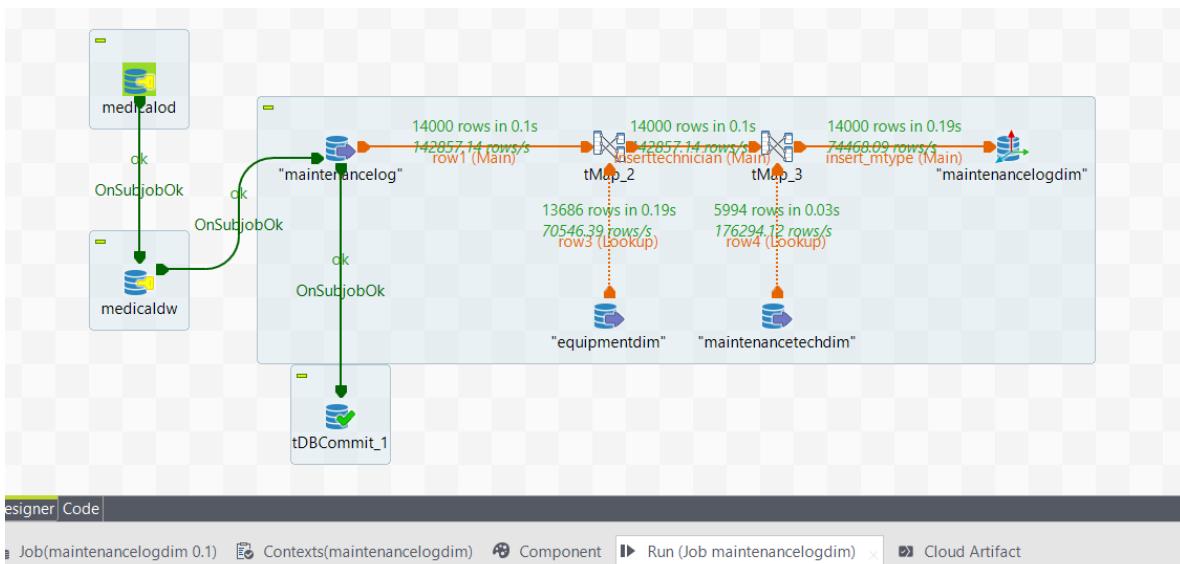
Column	K..	Type	N..	Date Pattern (Ctrl+D)	Length	Precision	Default	Comment
itemdescription	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		53	0		
partdescription	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		52	0		

Column	K..	Type	N..	Date Pattern (Ctrl+D)	Length	Precision	Default	Comment
itemkey	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		10	0		
itemdescription	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		
partdescription	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		

Data preview table:

	itemkey	itemdescription	partdescription
1	1	Batteries/ UPS	BATTERY LITHIUM ION
2	2	Sensors/ Detectors	SPO finger sensor mindray adult 512F
3	3	Keyboards/ Keypads/ Remotes	Foil Keyboard BBRAUN
4	4	Equipment Specific Parts	FISHER & PAYKEL ELECTRICAL ADAPTOR 900MR858
5	5	Solenoids/ Valves/ Gauges	Solenoid Valve (Chamber)
6	6	Equipment Specific Parts	EXPIRATORY CASSETTE
7	7	Diaphragms/ Pumps	Fresenius 4008S Concentrate Pump Assy Complete

11. Maintenance Logdim:



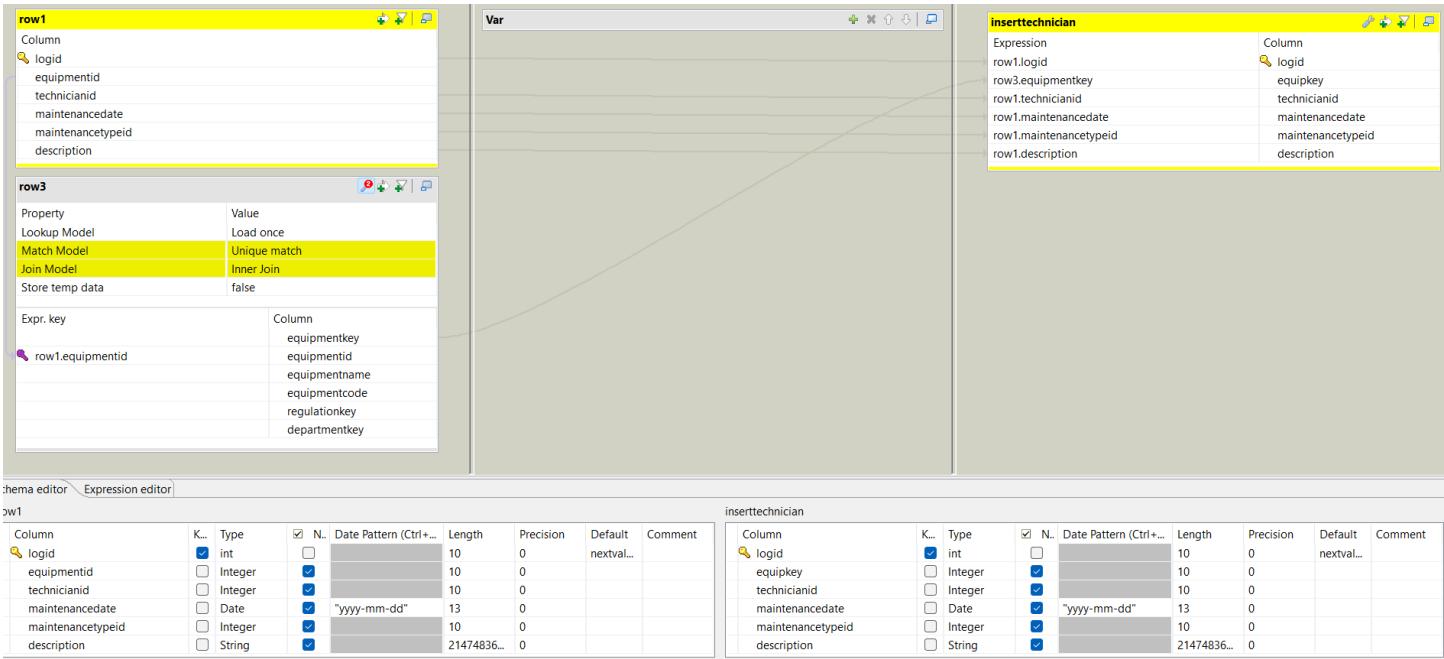
Job maintenance logdim execution details:

Execution tab (Run button selected):

Job maintenance logdim started at 18:46 15/11/2023. [statistics] connecting to socket on port 3374 [statistics] connected [statistics] disconnected

Job maintenance logdim ended at 18:46 15/11/2023. [Exit code = 0]

T-map1:



SCD Type3 :

The screenshot shows the SCD component editor with several tabs:

- Unused:** Contains the historicalmaintenancetypeid field.
- Type 0 fields:** Contains the equipkey and maintenancedate fields.
- Type 1 fields:** Contains the description field.
- Type 2 fields:** An empty tab.
- Versioning:** A table with columns: type, name, creation, and compleme... (partially visible). It contains rows for start (scd_start), end (scd_end), versi... (scd_version), and active (scd_active).
- Source keys:** Contains the logid field.
- Surrogate keys:** Contains fields for name (logkey), creation (Table max + 1), and complement.
- Type 3 fields:** A table with columns: current value, previous value, and historicalmaintenancetype... (partially visible). It contains rows for maintenancetypeid (historicalmaintenancetype...).

	logkey	logid	equipkey	maintenancetechid	maintainedate	maintenancetypeid	description
1	1	1	3,248	382	2023-09-15	5	Routine inspection and cleaning
2	2	2	4,647	4,331	2023-01-23	2	Disinfection and sterilization
3	3	3	5,291	3,824	2023-11-12	9	Routine inspection and cleaning
4	4	4	101	858	2023-10-07	9	Routine inspection and cleaning
5	5	5	1,385	1,849	2023-01-26	3	Replaced worn-out components
6	6	6	2,119	3,374	2023-07-13	3	Routine inspection and cleaning
7	7	7	4,280	4,042	2023-10-28	2	Disinfection and sterilization
8	8	8	5,812	2,237	2023-08-08	7	Diagnostic tests and troubleshooting
9	9	9	3,284	475	2023-02-25	5	Safety inspection and compliance check
10	10	10	4,001	1,879	2023-05-03	6	Safety inspection and compliance check
11	11	11	2,542	2,831	2023-09-17	2	Replaced worn-out components
12	12	12	2,863	4,388	2023-06-02	2	Verified calibration accuracy
13	13	13	1,726	3,174	2023-02-23	9	Calibrated equipment for accuracy
14	14	14	5,849	2,508	2023-02-21	6	Preventive maintenance
15	15	15	4,910	3,609	2023-03-10	8	Replaced worn-out components
16	16	16	5,170	1,982	2023-11-23	4	Verified calibration accuracy
17	17	17	2,260	2,211	2023-07-15	2	Verified calibration accuracy

12. Maintanencetypedim :

The screenshot shows the Talend Data Integration environment for a job named "Job(maintenancetypedim 0.1)".

Job Structure:

```

graph TD
    mediclod[mediclod] --> OnSubjobOk_1[OnSubjobOk]
    OnSubjobOk_1 --> medicaldw[medicaldw]
    medicaldw --> OnSubjobOk_2[OnSubjobOk]
    OnSubjobOk_2 --> maintenancetype["maintenancetype"]
    maintenancetype --> OnSubjobOk_3[OnSubjobOk]
    OnSubjobOk_3 --> tDBCommit_1[tDBCommit_1]
    tDBCommit_1 --> maintenancetypedim["maintenancetypedim"]
    
```

Job Log:

Starting job maintenancetypedim at 18:47 15/11/2023.
[statistics] connecting to socket on port 3543
[statistics] connected
[statistics] disconnected
Job maintenancetypedim ended at 18:47 15/11/2023. [Exit code = 0]

Job Results:

Column	Count
maintenancetypeid	10 rows
maintenancename	10 rows
description	10 rows
maintenancecost	10 rows

Component Details:

- mediclod:** A flat file source component.
- OnSubjobOk:** Subjob completion triggers.
- medicaldw:** A flat file target component.
- tDBCommit_1:** A database commit component.
- maintenancetype:** A database source component.
- tMap_1:** A mapping component with the following statistics:
 - 10 rows in 0.03s
 - 384.62 rows/s
 - insert (Main)
 - 10 rows in 0.05s
 - 188.68 rows/s
 - insert (Main)
- maintenancetypedim:** A database target component.

Schema Editor:

The schema editor shows the structure of the "row1" table and the "insert" statement.

row1:

Column	Type	Length	Precision	Default	Comment
maintenancetypeid	int	10	0		
maintenancename	String	255	0		
description	String	21474836..	0		
maintenancecost	Integer	10	0		

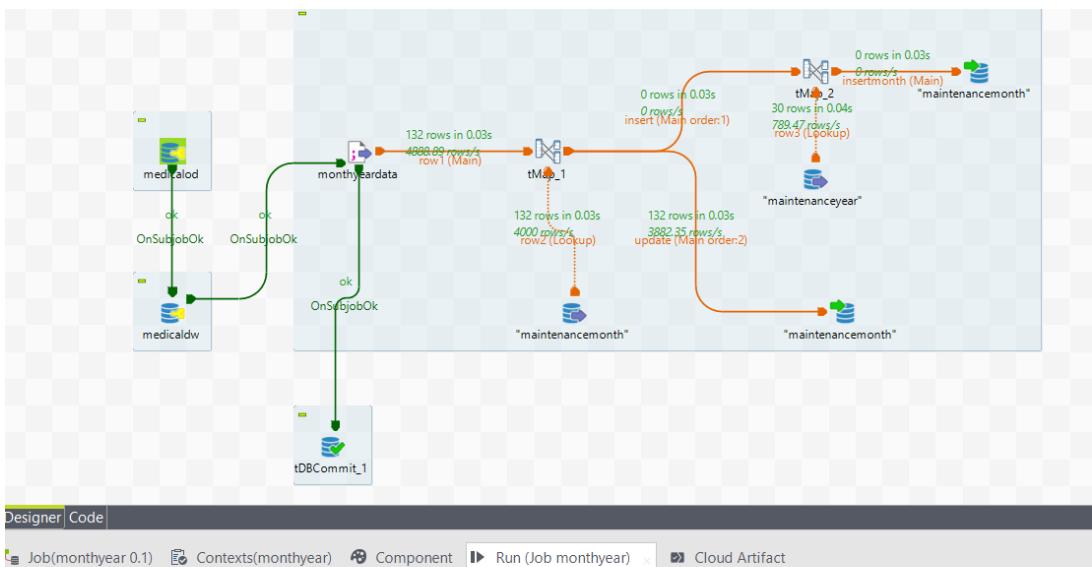
insert:

Column	Type	Length	Precision	Default	Comment
maintenancetypeid	int	10	0		
maintenancename	String	255	0		
description	String	21474836..	0		
maintenancecost	Integer	10	0		

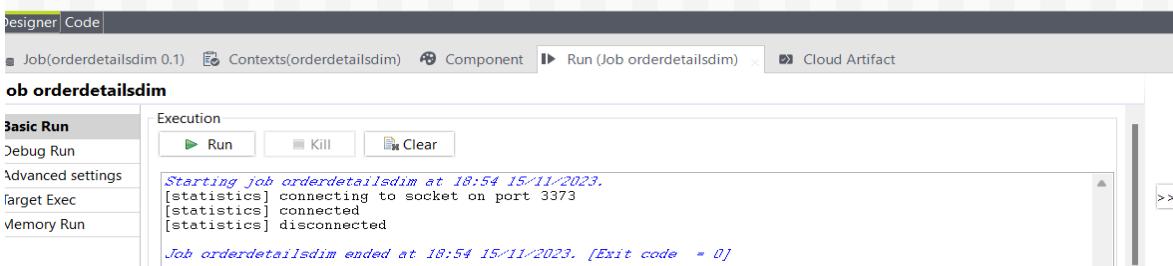
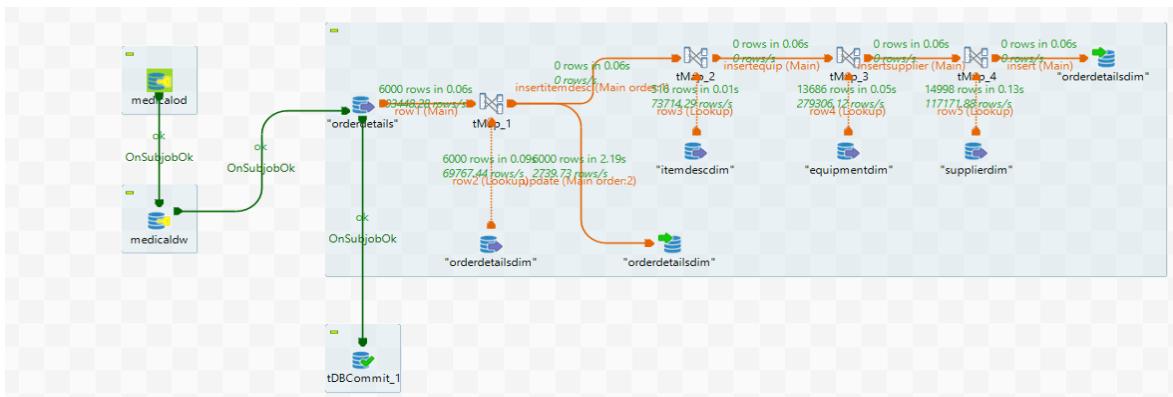
IE7374 Data Warehousing and Data Integration

123 maintenancetypekey	123 maintenancetypeid	ABC maintenancetypename	ABC description	123 maintenancecost	validfrom
1	1	Preventive Maintenance	Routine maintenance to prevent issues proactively.	500	2023-11-13
2	2	Corrective Maintenance	Repair or fix equipment when it malfunctions.	750	2023-11-13
3	3	Scheduled Maintenance	Planned maintenance based on a fixed schedule.	1,200	2023-11-13
4	4	Emergency Maintenance	Immediate maintenance for critical issues.	980	2023-11-13
5	5	Calibration	Adjusting and calibrating equipment for accuracy.	620	2023-11-13
6	6	Software Update	Updating equipment software for performance and security.	890	2023-11-13
7	7	Cleaning and Sanitization	Cleaning and sanitizing equipment regularly.	1,100	2023-11-13
8	8	Hardware Replacement	Replacing hardware components as needed.	950	2023-11-13

13. Monthyear:



14. Orderdetailsdim :



IE7374 Data Warehousing and Data Integration

t-map1:

The screenshot shows the Talend t-map1 interface with four main components:

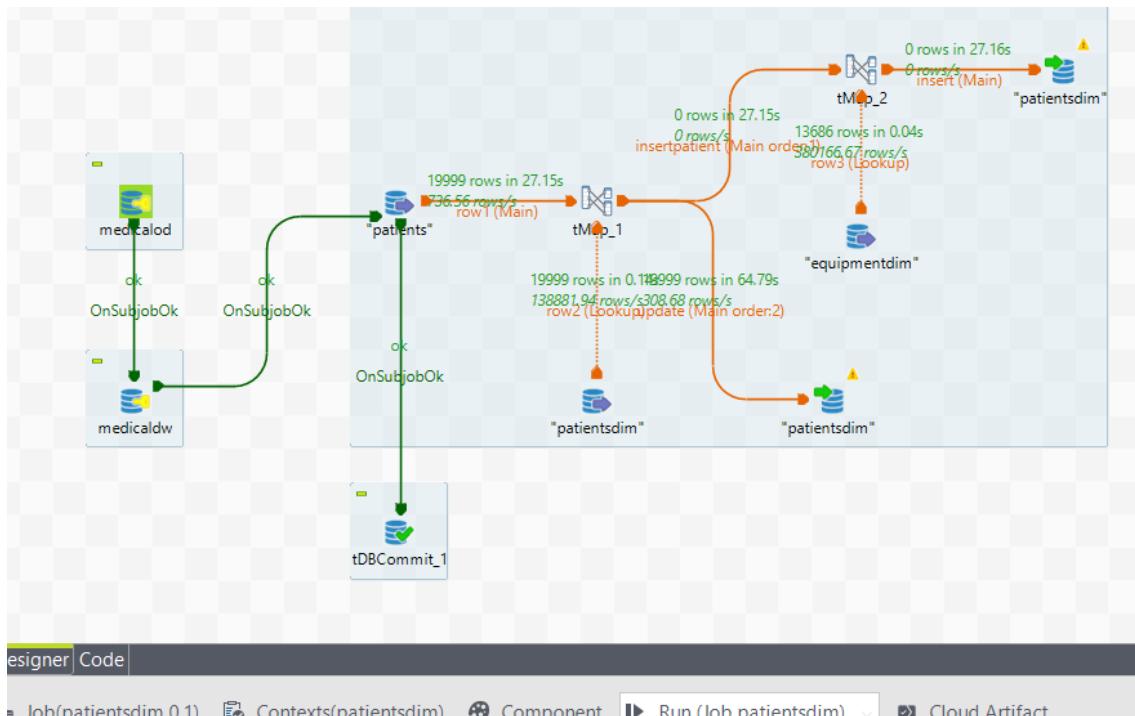
- row1**: A table with columns: Column,orderid, ordername, description, itemcode, itemdescription, partdescription, supplierid, equipmentid, totalcost, quantity.
- Var**: A table with a single column labeled "Var" containing many lines of code representing the data flow between components.
- insertitemdesc**: A table with columns: Property, Value. Properties include: Catch output reject (false), Catch lookup inner join reject (true), Schema Type (Built-In). Values correspond to the columns in row1.
- update**: A table with columns: Expression, Column. Expressions include: row2.orderdetailskey, row2.orderid, row2.ordername, row1.description, row2.itemcode. Columns correspond to the columns in insertitemdesc.

Below these components are two tables for schema definition:

- row1**: A table defining the schema for the input data.
- insertitemdesc**: A table defining the schema for the output data.

	123 orderdetailskey	123 orderid	abc ordername	abc description	abc itemcode	123 itemkey
1	224	224	Oximeter	Description for Oximeter	BP0025	50
2	348	347	Gas, Helium	Description for Gas, Helium	BP0004	51
3	2,322	2,319	Oximeter	Description for Oximeter	BP0004	51
4	3,150	3,150	Pneumotachometer	Description for Pneumotachometer	BP0006	51
5	4,103	4,103	Oximeter	Description for Oximeter	BP0004	51
6	4,134	4,134	Oximeter, Ear	Description for Oximeter, Ear	BP0025	50
7	4,541	4,539	Laryngoscope Kit	Description for Laryngoscope Kit	BP0007	40
8	2,038	2,036	Conserver, Oxygen	Description for Conserver, Oxygen	BP0004	51
9	1	1	Monitor, Carbon-Dioxide, Cutaneous	Description for Monitor, Carbon-Dioxide, Cutaneous	BP0018	41
10	2	2	Analyzer, Gas, Carbon-Monoxide, Gaseous-Phase	Description for Analyzer, Gas, Carbon-Monoxide, Gaseous-Phase	BP0025	50
11	3	3	Oximeter	Description for Oximeter	BP0005	51
12	4	4	Flowmeter, Calibration, Gas	Description for Flowmeter, Calibration, Gas	BP0001	50
13	5	5	Mask, Oxygen, Non-Rebreathing	Description for Mask, Oxygen, Non-Rebreathing	BP0025	50
14	6	6	Non-Bronchoscopic Bronchoalveolar Lavage Catheter	Description for Non-Bronchoscopic Bronchoalveolar Lavage	BP0025	50
15	7	7	Monitor, Oxygen, Cutaneous, For Uses Other Than For Infant	Description for Monitor, Oxygen, Cutaneous, For Uses Other Than For Infant	BP0003	50
16	8	8	Medevac Use Oxygen Generator	Description for Medevac Use Oxygen Generator	BP0001	50
17	9	9	Acid-Vanilmandelic-Electrophoretic Separation	Description for Acid-Vanilmandelic-Electrophoretic Separation	BP0011	50

15. Patientdim:



Job patientsdim

Basic Run
Debug Run
Advanced settings
target Exec
Memory Run

Execution

Starting job patientsdim at 18:57 15/11/2023.
[statistics] connecting to socket on port 3370
[statistics] connected
[statistics] disconnected

Job patientsdim ended at 18:58 15/11/2023. [Exit code = 0]

row1

Column	patientid	gender	agecategory	description	paymentamount	procedure	equipmentid
patientid							
gender							
agecategory							
description							
paymentamount							
procedure							
equipmentid							

row2

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Inner Join
Store temp data	false

Expr. key

Column	patientkey	patientid	gender	agecategory	description	paymentamount	procedure
row1.patientid							

Var

Insertpatient

Property	Value
Catch output reject	false
Catch lookup inner join reject	true
Schema Type	Built-In

Expression

Column	patientid	gender	agecategory	description	paymentamount	procedure	equipmentid
row1.patientid							
row1.gender							
row1.agecategory							
row1.description							
row1.paymentamount							
row1.procedure							
row1.equipmentid							

update

Expression	Column
row2.patientkey	patientkey
row2.patientid	patientid
row2.gender	gender
row2.agecategory	agecategory
row2.description	description
row2.paymentamount	paymentamount
row2.procedure	procedure
row2.equipmentid	equipmentid

schema editor Expression editor

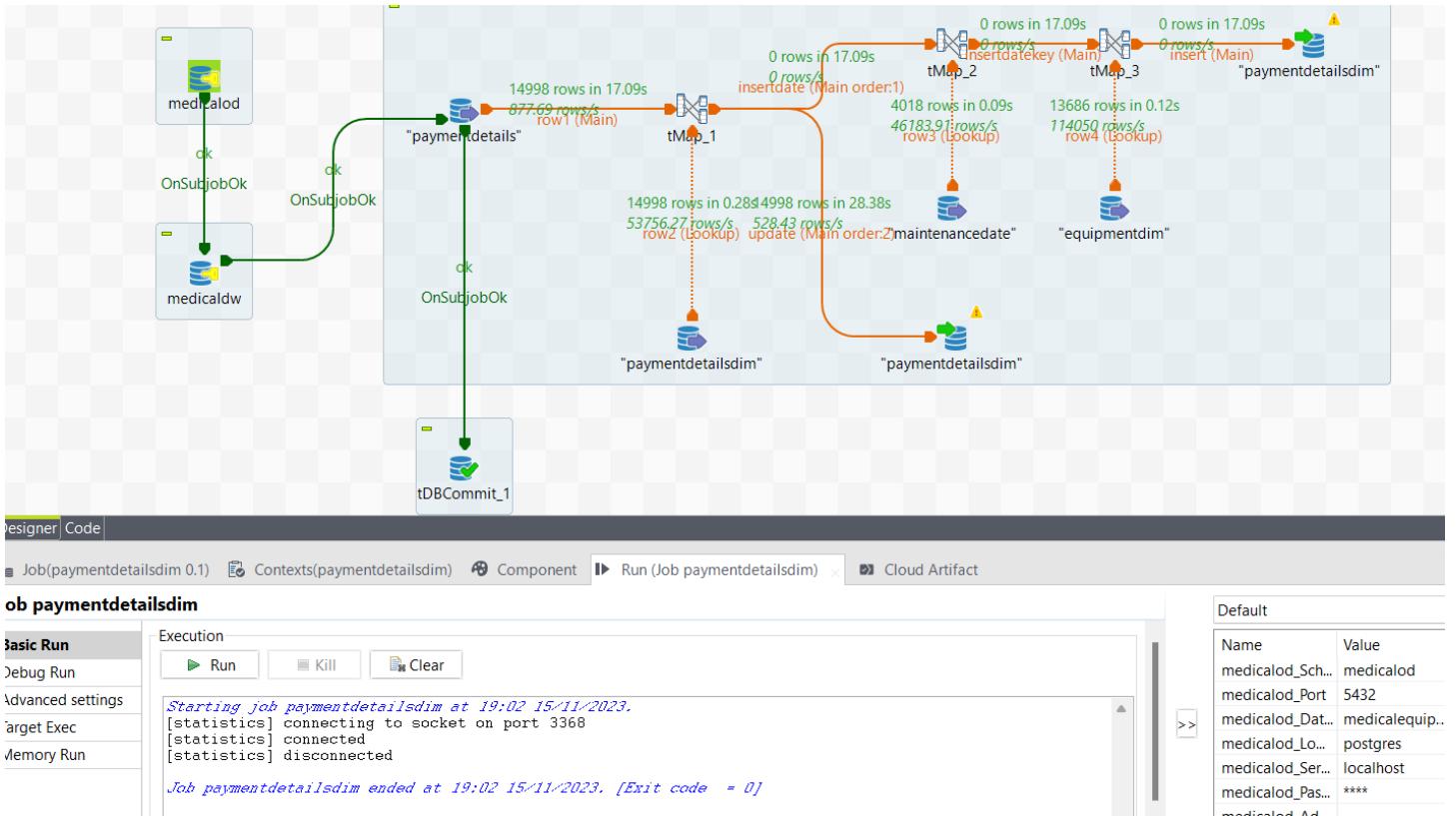
Column	K...	Type	N...	Date Pattern (Ctrl+...	Length	Precision	Default	Comment
patientid	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		10	0	nextval...	
gender	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		10	0		
agecategory	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		
description	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		21474836...	0		
paymentamount	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		10	0		
procedure	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		10	0		

Column	K...	Type	N...	Date Pattern (Ctrl+...	Length	Precision	Default	Comment
patientid	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		10	0		
gender	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		10	0		
agecategory	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		
description	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		21474836...	0		
paymentamount	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		10	0		
procedure	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		10	0		

IE7374 Data Warehousing and Data Integration

	<code>patientkey</code>	<code>patientid</code>	<code>gender</code>	<code>agecategory</code>	<code>description</code>	<code>paymentamount</code>	<code>procedure</code>	<code>eu</code>
1	191	191	Male	Under 25 years old	Standard wheelchair	40	6	
2	1,112	1,110	Male	Between 55-65 years old	Outflare wedge	40	2	
3	1,245	1,244	Male	Between 35-45 years old	Standard wheelchair	30	9	
4	1,888	1,887	Male	Under 25 years old	Standard wheelchair	30	5	
5	1,594	1,592	Male	Between 25-35 years old	Iv pole	[NULL]	1	
6	1,889	1,888	Male	Between 35-45 years old	Standard wheelchair	40	1	
7	1,890	1,889	Male	Under 25 years old	Lightweight wheelchair	80	3	

16. PaymentDetailsdim:



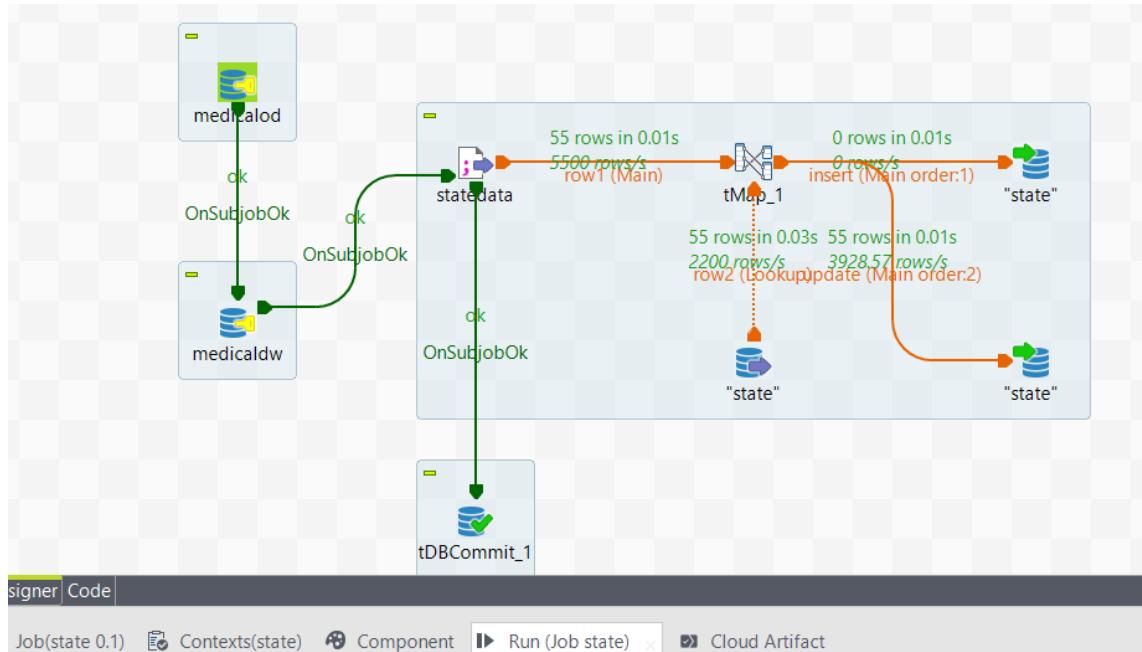
Job configuration details:

- row1:** Contains columns: paymentid, supplierid, equipmentid, amount, purchasedate, invoiceno, logid.
- row2:** Contains properties: Lookup Model (Load once), Match Model (Unique match), Join Model (Inner Join), Store temp data (false).
- Var:** Expression: `Math.round((row1.amount == null ? 0 : ...))`, Type: float, Variable: updatedamount.
- insertdate:** Property: Value: false; Catch lookup inner join reject: true; Schema Type: Built-In. Columns: paymentid, supplierid, equipmentid, amount, purchasedate, invoiceno, logid, updatedamount.
- update:** Expression: row2.paymentkey, Columns: paymentkey, paymentid, supplierid, equipmentid, amount.

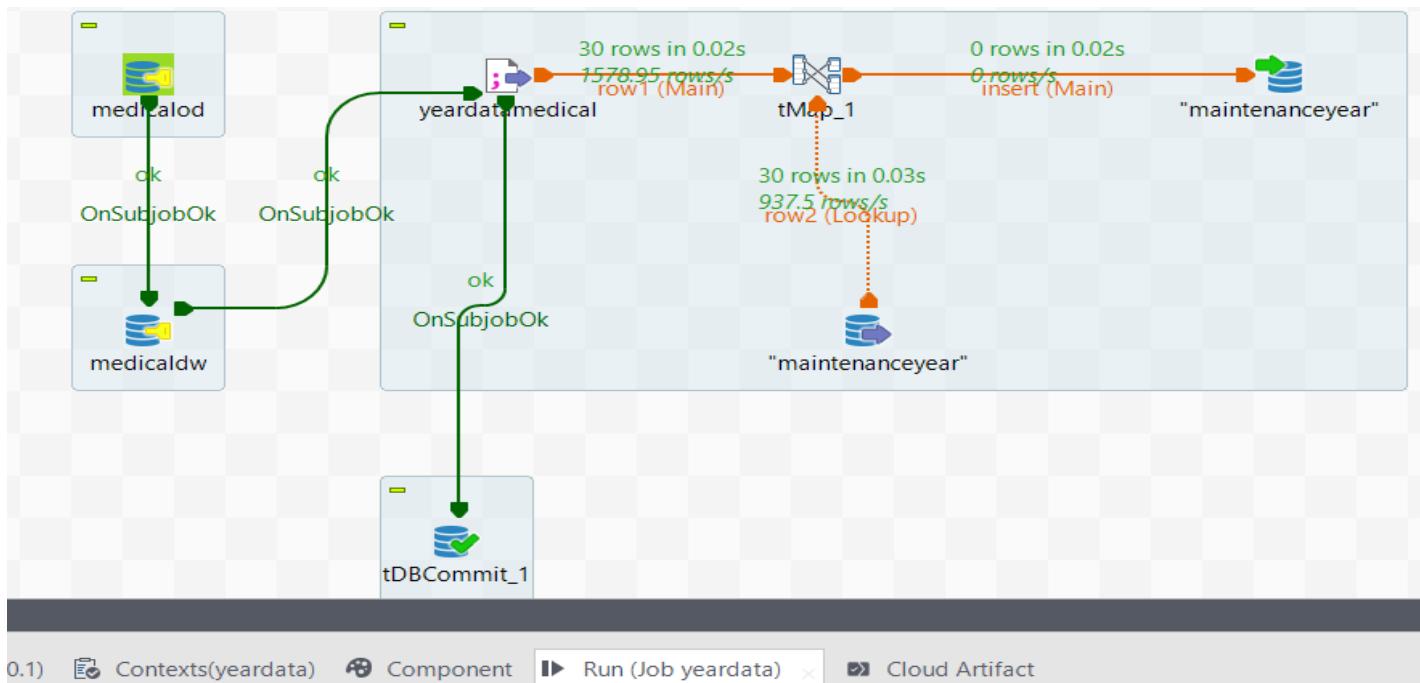
IE7374 Data Warehousing and Data Integration

	123 paymentkey	123 paymentid	123 supplierid	123 equipkey	123 logid	123 amount	123 discount	123 updatedamount	⌚ purchasedate	abc paymentde
1	7,508	7,508	7,508	2,631	1,507	[NULL]	5	0	2023-05-28	[NULL]
2	1	1	1	1,900	1	18,894.849609375	5	17,950	2023-06-12	[NULL]
3	2	2	2	3,616	2	2,831.0400390625	5	2,689	2023-06-16	[NULL]
4	3	3	3	5,192	3	14,667.990234375	5	13,935	2023-03-03	[NULL]
5	4	4	4	2,801	4	93,940.703125	5	89,244	2023-10-01	[NULL]
6	5	5	5	2,952	5	18,335.55078125	5	17,419	2023-02-05	[NULL]
7	6	6	6	5,596	6	3,948.6398925781	5	3,751	2023-10-11	[NULL]
8	7	7	7	3,434	7	56,000.96875	5	53,201	2023-06-01	[NULL]
9	8	8	8	3,367	8	9,062.5302734375	5	8,609	2023-08-25	[NULL]
10	9	9	9	4,341	9	6,507.8500976562	5	6,182	2023-10-24	[NULL]
11	10	10	10	5,494	10	301,050.59375	5	285,998	2023-06-25	[NULL]
12	11	11	11	2,080	11	120,359.2265625	5	114,341	2023-02-04	[NULL]
13	12	12	12	4,404	12	51,484.69921875	5	48,910	2023-04-06	[NULL]
14	13	13	13	1,256	13	10,400.7001953125	5	9,881	2023-04-27	[NULL]

17. State :



18. Yeardata:



0.1) Contexts(yeardata) Component Run (Job yeardata) Cloud Artifact

Execution

Run Kill Clear

```

Starting job yeardata at 19:14 15/11/2023.
[statistics] connecting to socket on port 3515
[statistics] connected
[statistics] disconnected
Job yeardata ended at 19:14 15/11/2023. [Exit code = 0]

```

The screenshot shows the Kettle Transformation Editor interface with three panels:

- Row1:** A table with one column "Year".
- Row2:** A table showing properties for a join operation:

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Inner Join
Store temp data	false

 A row mapping section shows "Expr. key" (row1.Year) mapped to "Column" (year).
- Var:** An empty variable editor panel.
- insert:** A configuration panel for an insert operation:

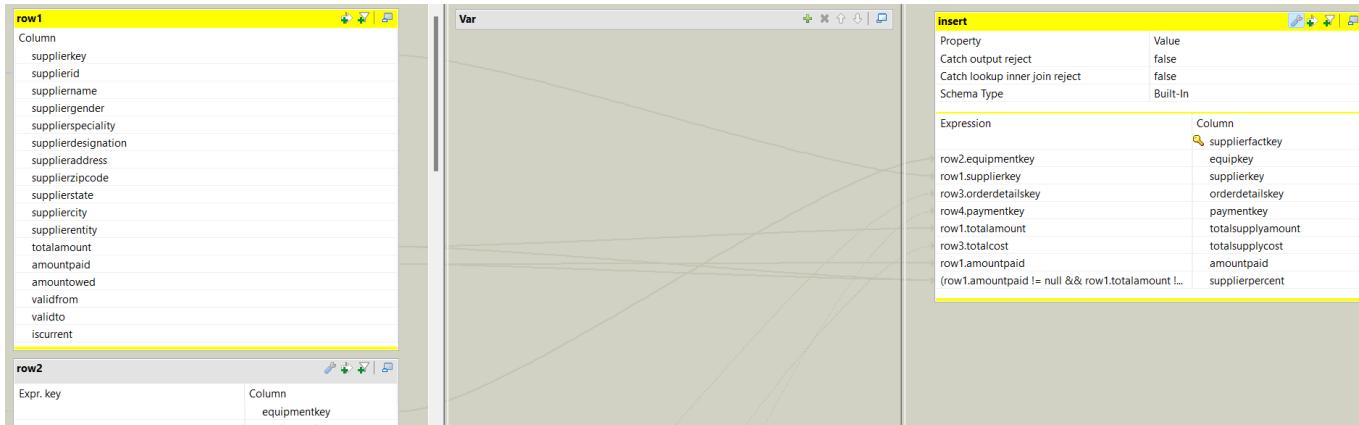
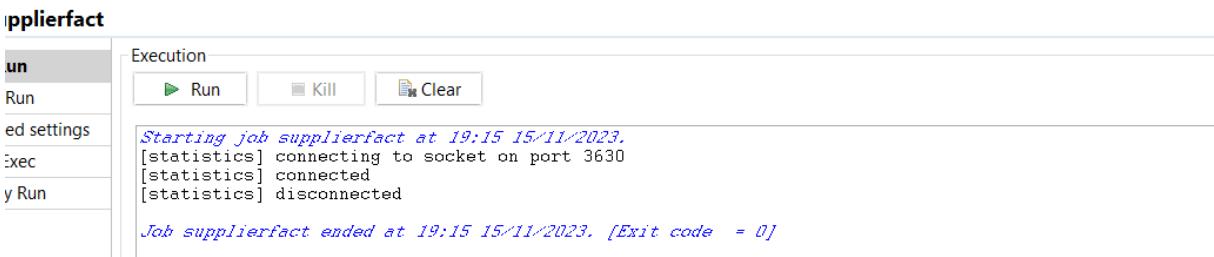
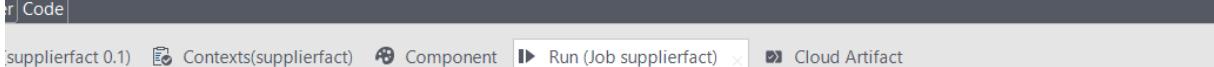
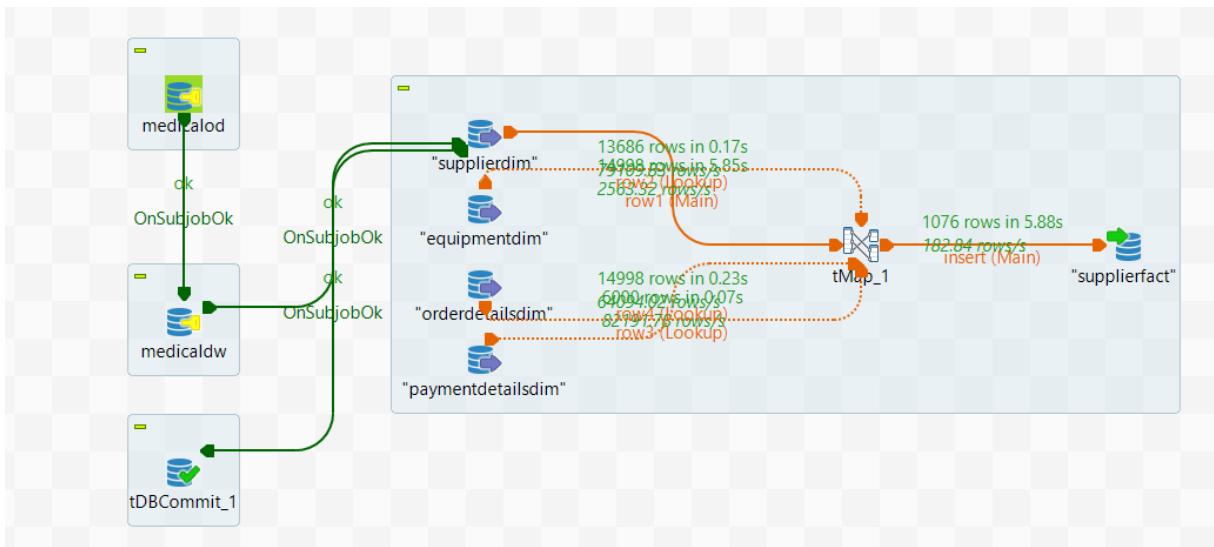
Property	Value
Catch output reject	false
Catch lookup inner join reject	true
Schema Type	Built-In

 An expression "row1.Year" is mapped to a column "year".
- update:** A configuration panel for an update operation:

Property	Value
Expression	row2.year

 An expression "row2.year" is mapped to a column "year".

19. SupplierFACT:



	123 supplierfactkey	123 equipkey	123 supplierkey	123 orderdetailskey	123 paymentkey	123 totalsupplyamount	123 totalsupplycost	123 amountpaid	123 supplier
1	738448	1,900	1	5,922	1	18,894,849,609,375	1,524,726,5625	11,850,370,117,1875	62
2	738449	1,900	1	5,922	1	18,894,849,609,375	1,524,726,5625	11,850,370,117,1875	62
3	738450	3,616	2	5,923	2	2,831,040,039,0625	3,561,640,380,8594	1,581,739,990,2344	55
4	738451	3,616	2	5,923	2	2,831,040,039,0625	3,561,640,380,8594	1,581,739,990,2344	55
5	738452	5,192	3	5,924	3	14,667,990,234,375	2,519,426,513,6719	6,828,850,097,6562	41
6	738453	5,192	3	5,924	3	14,667,990,234,375	2,519,426,513,6719	6,828,850,097,6562	41
7	738454	2,801	4	5,925	4	93,940,703,125	3,027,724,121,0938	52,868,320,3125	56
8	738455	2,801	4	5,925	4	93,940,703,125	3,027,724,121,0938	52,868,320,3125	56
9	738456	2,952	5	5,926	5	18,335,550,78125	2,630,599,853,156	10,021,309,570,3125	54
10	738457	2,952	5	5,926	5	18,335,550,78125	2,630,599,853,156	10,021,309,570,3125	54
11	738458	5,596	6	5,927	6	3,948,639,892,5781	4,686,276,855,4688	2,090,320,068,3594	52
12	738459	5,596	6	5,927	6	3,948,639,892,5781	4,686,276,855,4688	2,090,320,068,3594	52
13	738460	3,434	7	5,928	7	56,000,96875	753,423,828,125	26,327,150,390,625	47
14	738461	3,434	7	5,928	7	56,000,96875	753,423,828,125	26,327,150,390,625	47
15	738462	3,367	8	5,929	8	9,062,530,273,4375	207,137,222,29	5,941,930,175,7812	65
16	738463	3,367	8	5,929	8	9,062,530,273,4375	207,137,222,29	5,941,930,175,7812	65
17	738464	4,341	9	5,930	9	6,507,850,097,6562	2,357,435,302,7344	2,458,580,078,125	37

Analytical Queries:

1. Top 3 paid maintenance Technicians:

```
/* to find the technicians who are highest paid */

select m.fullname, m3.description, sum(m3.maintenancecost) as amount
from medicaldw.maintenancetechdim m inner join
medicaldw.maintenancelogdim m2 on m.maintenancetechkey = m2.maintenancetechid
inner join medicaldw.maintenancetypedim m3 on m.maintenancetechkey = m3.maintenancetypekey
group by m.fullname , m3.description
order by amount desc limit 3
```

maintenancetechdim(+) 1 ×

select m.fullname, m3.description, sum(m3.mainte | Enter a SQL expression to filter results (use Ctrl+Space)

	ABC fullname	ABC description	123 amount
1	Harry Hayes	Replacing hardware components as needed.	4,750
2	Calvin Coolidge	Upgrading equipment for improved functionality.	2,880
3	Ulysses Buchanan	Regular inspections to identify potential issues.	2,700

2. List user access levels and the access they have.

```
/*2. List user access levels and the access they have. */
select u.username, u.role
from medicaldw.useraccessdim u
inner join medicaldw.supplierdim s
on u.userid = s.supplierid
```

useraccessdim 1 ×

select username, u.role from medicaldw.useracc | Enter a SQL expression to filter results (use Ctrl+Space)

	ABC username	ABC role
89	ZANA CORREA	Owner
90	RAYMOND BARROWS	Supplier
91	WILLAIM GRABENSTEIN	Owner
92	MARY OSTASZEWSKI	Supplier
93	TIFFANY BERRY	Owner
94	MICHAEL OSLEBER	Supplier
95	FRANK HUX	Owner
96	ERIK RATCHFORD	Supplier
97	TREAH HAGGERTY	Owner
98	EMILY BYRNE	Supplier

3. Total maintenance cost to the healthcare/hospitals:

The screenshot shows a data warehousing tool interface with a SQL editor and a results grid.

```
/* 3. Total maintainence cost */
select e.equipmentname , m.maintenancetypename , sum(m.maintenancecost)
from medicaldw.maintenancetypedim m
inner join medicaldw.maintenancelogdim m2
on m.maintenancetypeid = m2.maintenancetypeid
inner join medicaldw.equipmentdim e on
e.equipmentkey = m2.equipkey
group by 1,2
```

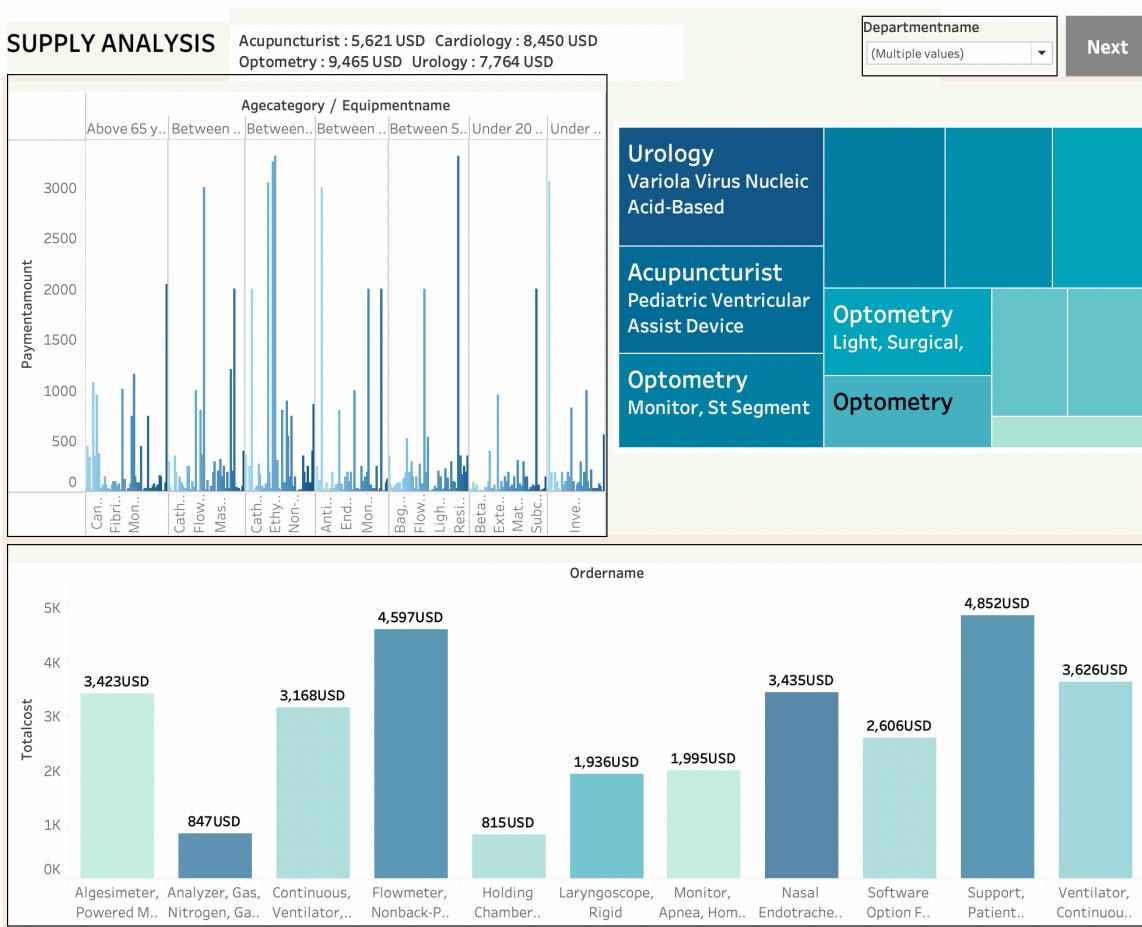
Grid results:

	equipmentname	maintenancetypename	sum
1	1-Nitroso-2-Naphthol (Fluorometric), Free Tyrosine	Calibration	2,480
2	11-Dehydro Thromboxane B2 Kit, Urinary	Scheduled Maintenance	2,400
3	2,4-Dinitrofluorobenzene (Spectroscopic), Nitrogen (Amino-Nitrogen)	Emergency Maintenance	1,960
4	2,4-Dinitrophenylhydrazine, Lactate Dehydrogenase	Hardware Replacement	1,900
5	2,4-Dinitrophenylhydrazine, Lactate Dehydrogenase	Software Update	1,780
6	25-OH-Vitamin D Mass Spectrometry Test System	Software Update	1,780
7	5-Amp-Phosphate Release (Colorimetric Test), 5'-Nucleotidase	Hardware Replacement	1,900
8	A Chemical Vapor Sterilization Multivariable Chemical Indicator	Cleaning and Sanitization	2,200
9	A Chemical Vapor Sterilization Multivariable Chemical Indicator	Preventive Maintenance	2,000
10	Aberrometer, Ophthalmic	Corrective Maintenance	1,500
11	Aberrometer, Ophthalmic	Emergency Maintenance	1,960
12	Aberrometer, Ophthalmic	Scheduled Maintenance	2,400
13	Ablation System, High Intensity Focused Ultrasound (Hifu), Mr-Guided	Corrective Maintenance	1,500

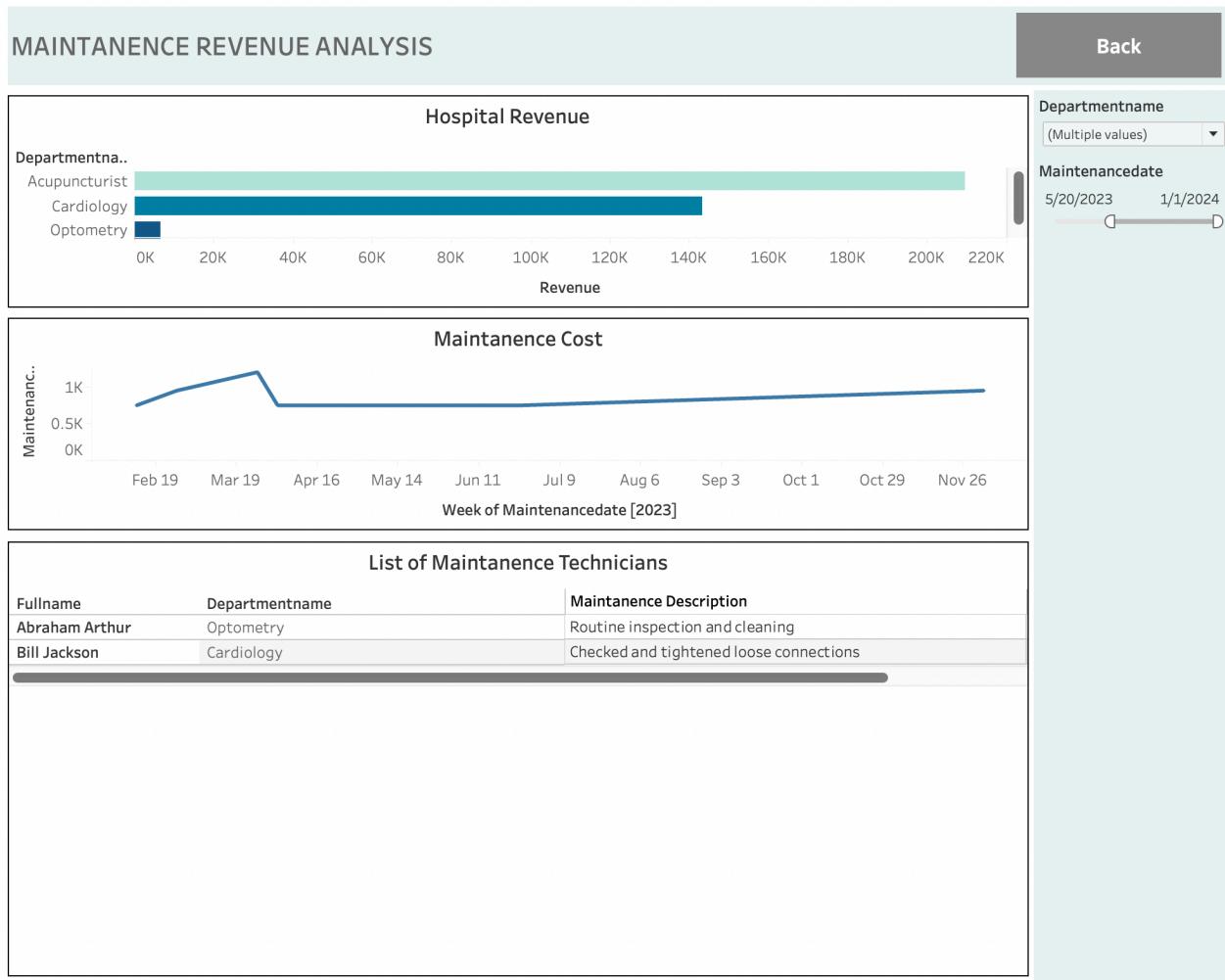
Data Analysis (KPI metrics): (Dashboards)

Supply Revenue Analysis:

We examine in our analysis the money made and expenses related to purchasing equipment for several departments in a healthcare facility. First, tree map that shows how much equipment each department has purchased in spatial terms. After that, on taking a closer look at the expenses the hospital must pay for the purchase of equipment while taking into account the particular requirements of every department, we notice certain insights that can help analyze the fund usage.



Furthermore, here, we notice how these newly purchased items of equipment are used and the related expenses that patients, broken down by age groups. This enables us to comprehend how the use of equipment affects various groups of people. We determine the overall cost allocated to each department to give a thorough overview and to provide important financial context for the acquisition and use of equipment across the healthcare institution.

Maintenance Revenue Analysis :

Determining how hospital money are allocated to equipment maintenance is the major goal of this maintenance revenue study. First, we figure out how much of the hospital's money goes toward keeping certain pieces of equipment in working order. We also perform a thorough analysis of the professionals performing various maintenance jobs to clarify their roles and duties inside the facility.

Additionally, our analysis looks at how maintenance expenses vary over the course of several weeks. Our objective is to evaluate the fluctuations in maintenance expenses over time to spot any patterns or trends that could affect the allocation of resources for continuous equipment maintenance and budgetary concerns. This all-encompassing method offers a thorough grasp of the financial and operational facets related to hospital equipment maintenance.