Medical inventory optimization

Pre-processing Code (SQL) by Rohit Paul

Software: MySQL Workbench

1. Creating the database.

```
CREATE DATABASE med_inventory;
```

2. Set the current database to "med_inventory".

```
USE med_inventory;
```

3. Importing the table in CSV format into MySQL.

Right-click on tables -> table data import wizard -> browse to select the table ‘projectfinaldata’.

4. Displaying the table.

```
SELECT * FROM projectfinaldata LIMIT 20;
```

Output:

Typeofsales	Patient_ID	Specialisation	Dept	Dateofbill	Quantity	ReturnQuantity	Final_Cost	Final_Sales	RtnMRP	Formulation	DrugName	SubCat	SubCat1
Sale	12018098765	Specialisation6	Department1	6-1-2022	1	0	55.406	59.26	0	Form1	ZINC ACETATE 20MG/5ML SYP	SYRUP & SUSPENSION	VITAMINS & MINERALS
Sale	12018103897	Specialisation7	Department1	7/23/2022	1	0	768.638	950.8	0	Form1	CEFTAZIDIME 2GM+AVIBACTAM 500MG	INJECTIONS	ANTI-INFECTIVES
Sale	12018101123	Specialisation2	Department3	6/23/2022	1	0	774.266	4004.214	0	Form2	EPTIFIBATIDE 0.75MG/ML	INJECTIONS	CARDIOVASCULAR & HEMATOPOIETIC SYSTEM
Sale	12018079281	Specialisation40	Department1	3/17/2022	2	0	40.798	81.044	0	Form1	WATER FOR INJECTION 10ML SOLUTION	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018117928	Specialisation5	Department1	12/21/2022	1	0	40.434	40.504	0	Form1	LORAZEPAM 1MG	TABLETS & CAPSULES	CENTRAL NERVOUS SYSTEM
Return	12018103662	Specialisation2	Department1	7/15/2022	0	8	47.902	0	330	Form1	SALBUTAMOL 2.5MG	INHALERS & RESPULES	RESPIRATORY SYSTEM
Sale	12018097585	Specialisation2	Department1	5/22/2022	1	0	41.862	42.218	0	Form1	FUROSEMIDE 10MG/ML	INJECTIONS	CARDIOVASCULAR & HEMATOPOIETIC SYSTEM
Sale	12018077721	Specialisation4	Department1	1-12-2022	3	0	60.026	142.752	0	Form1	SODIUM CHLORIDE IVF 100ML	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018096500	Specialisation4	Department2	8/24/2022	2	0	49.856	94	0	Form2	SODIUM BICARBONATE 8.5% INJ	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018071649	Specialisation4	Department1	8/31/2022	1	0	258.86	319.8	0	Form1	PEPTIDE BASED DIET POWDER	NUTRITIONAL SUPPLEMENTS	NUTRITION
Sale	12018074894	Specialisation7	Department1	10-4-2022	3	0	114.592	290.4	0	Form1	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018088348	Specialisation4	Department1	4-2-2022	20	0	231.834	1294	0	Form1	N-ACETYLCYSTEINE 1000MG/5ML INJ	INJECTIONS	RESPIRATORY SYSTEM
Sale	12018101319	Specialisation16	Department2	7-1-2022	1	0	66.88	102.6	0	Form1	PROPOFOL 1% 20ML INJ	INJECTIONS	ANAESTHETICS
Sale	12018108547	Specialisation6	Department1	8/20/2022	8	0	52.204	343.84	0	Form1	PARACETAMOL 150MG	INJECTIONS	CENTRAL NERVOUS SYSTEM
Sale	12018080245	Specialisation7	Department1	7/29/2022	1	0	41.658	43.2	0	Form1	VITAMIN K 1ML INJ	INJECTIONS	CARDIOVASCULAR & HEMATOPOIETIC SYSTEM
Sale	12018115496	Specialisation2	Department1	11/26/2022	2	0	89.728	193.6	0	Form1	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018111286	Specialisation25	Department1	9-7-2022	1	0	49.352	60.8	0	Form1			
Sale	12018097033	Specialisation20	Department1	9/17/2022	2	0	40.34	81.1	0	Form1			
Return	12018122962	Specialisation54	Department1	12/19/2022	0	2	70.016	0	115	Form2	SODIUM CHLORIDE 0.9%	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018106746	Specialisation20	Department1	10/14/2022	2	0	40.34	81.1	0	Form1			

Observation: The 'Dateofbill' column in the dataset exhibits inconsistent date formats, with a combination of forward slashes ("/") and dashes ("-"). To ensure uniformity, we can store dates in a consistent format by replacing the forward slashes with dashes.

5. Checking the schema of the dataset to ensure that all columns have the correct format.

DESCRIBE projectfinaldata;

Output:

Field	Type	Null	Key	Default	Extra
Typeofsales	text	YES		NULL	
Patient_ID	bigint	YES		NULL	
Specialisation	text	YES		NULL	
Dept	text	YES		NULL	
Dateofbill	text	YES		NULL	
Quantity	int	YES		NULL	
ReturnQuantity	int	YES		NULL	
Final_Cost	double	YES		NULL	
Final_Sales	double	YES		NULL	
RtnMRP	int	YES		NULL	
Formulation	text	YES		NULL	
DrugName	text	YES		NULL	
SubCat	text	YES		NULL	
SubCat1	text	YES		NULL	

Observation: The 'Dateofbill' column in the dataset is currently set as "text", but it would be better to store dates in a date-specific data type like "DATE" to ensure proper handling and sorting of dates.

6. Fixing inconsistent date formats in the Dateofbill column and creating a new table clean_projectfinaldata with transformed date values and the other selected columns from the 'projectfinaldata' table.

CREATE TABLE clean_projectfinaldata AS

SELECT Typeofsales, Patient_ID, Specialisation, Dept,

STR_TO_DATE(REPLACE(Dateofbill,'/','-'),'%m-%d-%Y') AS Dateofbill, Quantity, ReturnQuantity,

Final_Cost, Final_Sales, RtnMRP, Formulation, DrugName, SubCat, SubCat1

FROM projectfinaldata;

Datatype and format of 'Dateofbill' column after Formatting:

Field	Type	Null	Key	Default	Extra
Typeofsales	text	YES		NULL	
Patient_ID	bigint	YES		NULL	
Specialisation	text	YES		NULL	
Dept	text	YES		NULL	
Dateofbill	date	YES		NULL	
Quantity	int	YES		NULL	
ReturnQuantity	int	YES		NULL	
Final_Cost	double	YES		NULL	
Final_Sales	double	YES		NULL	

Typeofsales	Patient_ID	Specialisation	Dept	Dateofbill	Quantity
Sale	12018098765	Specialisation6	Department1	2022-06-01	1
Sale	12018103897	Specialisation7	Department1	2022-07-23	1
Sale	12018101123	Specialisation2	Department3	2022-06-23	1
Sale	12018079281	Specialisation40	Department1	2022-03-17	2
Sale	12018117928	Specialisation5	Department1	2022-12-21	1
Return	12018103662	Specialisation2	Department1	2022-07-15	0
Sale	12018097585	Specialisation2	Department1	2022-05-22	1
Sale	12018077721	Specialisation4	Department1	2022-01-12	3
Sale	12018096500	Specialisation4	Department2	2022-08-24	2

7. Counting the missing and non-missing values for each column and the total number of rows in the 'clean_projectfinaldata' table.

```
SELECT
  COUNT(CASE WHEN TRIM(Typeofsales) = '' OR Typeofsales IS NULL THEN 1 END) AS typeofsales_missing,
  COUNT(CASE WHEN TRIM(Typeofsales) <> '' AND Typeofsales IS NOT NULL THEN 1 END) AS typeofsales_non_missing,
  COUNT(CASE WHEN Patient_ID IS NULL THEN 1 END) AS patient_id_missing,
  COUNT(CASE WHEN Patient_ID IS NOT NULL THEN 1 END) AS patient_id_non_missing,
  COUNT(CASE WHEN TRIM(Specialisation) = '' OR Specialisation IS NULL THEN 1 END) AS specialisation_missing,
  COUNT(CASE WHEN TRIM(Specialisation) <> '' AND Specialisation IS NOT NULL THEN 1 END) AS specialisation_non_missing,
  COUNT(CASE WHEN TRIM(Dept) = '' OR Dept IS NULL THEN 1 END) AS dept_missing,
  COUNT(CASE WHEN TRIM(Dept) <> '' AND Dept IS NOT NULL THEN 1 END) AS dept_non_missing,
  COUNT(CASE WHEN TRIM(Dateofbill) = '' OR Dateofbill IS NULL THEN 1 END) AS dateofbill_missing,
  COUNT(CASE WHEN TRIM(Dateofbill) <> '' AND Dateofbill IS NOT NULL THEN 1 END) AS dateofbill_non_missing,
  COUNT(CASE WHEN Quantity IS NULL THEN 1 END) AS quantity_missing,
  COUNT(CASE WHEN Quantity IS NOT NULL THEN 1 END) AS quantity_non_missing,
  COUNT(CASE WHEN ReturnQuantity IS NULL THEN 1 END) AS returnquantity_missing,
  COUNT(CASE WHEN ReturnQuantity IS NOT NULL THEN 1 END) AS returnquantity_non_missing,
  COUNT(CASE WHEN Final_Cost IS NULL THEN 1 END) AS final_cost_missing,
  COUNT(CASE WHEN Final_Cost IS NOT NULL THEN 1 END) AS final_cost_non_missing,
  COUNT(CASE WHEN Final_Sales IS NULL THEN 1 END) AS final_sales_missing,
  COUNT(CASE WHEN Final_Sales IS NOT NULL THEN 1 END) AS final_sales_non_missing,
  COUNT(CASE WHEN RtnMRP IS NULL THEN 1 END) AS rtnmrp_missing,
  COUNT(CASE WHEN RtnMRP IS NOT NULL THEN 1 END) AS rtnmrp_non_missing,
  COUNT(CASE WHEN TRIM(Formulation) = '' OR Formulation IS NULL THEN 1 END) AS formulation_missing,
  COUNT(CASE WHEN TRIM(Formulation) <> '' AND Formulation IS NOT NULL THEN 1 END) AS formulation_non_missing,
  COUNT(CASE WHEN TRIM(DrugName) = '' OR DrugName IS NULL THEN 1 END) AS drugname_missing,
```

```
  COUNT(CASE WHEN TRIM(DrugName) <> '' AND DrugName IS NOT NULL THEN 1 END) AS drugname_non_missing,

  COUNT(CASE WHEN TRIM(SubCat) = '' OR SubCat IS NULL THEN 1 END) AS subcat_missing,

  COUNT(CASE WHEN TRIM(SubCat) <> '' AND SubCat IS NOT NULL THEN 1 END) AS subcat_non_missing,

  COUNT(CASE WHEN TRIM(SubCat1) = '' OR SubCat1 IS NULL THEN 1 END) AS subcat1_missing,

  COUNT(CASE WHEN TRIM(SubCat1) <> '' AND SubCat1 IS NOT NULL THEN 1 END) AS subcat1_non_missing,

  COUNT(*) AS total_rows

FROM clean_projectfinaldata;
```

Output:

typeofsales_missing	typeofsales_non_missing	patient_id_missing	patient_id_non_missing	specialisation_missing	specialisation_non_missing	dept_missing	dept_non_missing	dateofbill_missing	dateofbill_non_missing
0	14218	0	14218	0	14218	0	14218	0	14218

quantity_missing	quantity_non_missing	returnquantity_missing	returnquantity_non_missing	final_cost_missing	final_cost_non_missing	final_sales_missing	final_sales_non_missing	rtnmrp_missing	rtnmrp_non_missing
0	14218	0	14218	0	14218	0	14218	0	14218

formulation_missing	formulation_non_missing	drugname_missing	drugname_non_missing	subcat_missing	subcat_non_missing	subcat1_missing	subcat1_non_missing	total_rows
653	13565	1668	12550	1668	12550	1692	12526	14218

Observation:

Missing values have been identified in the following columns: Formulation, DrugName, SubCat, and SubCat1.

8. Replacing the missing values with ‘unknown’ in the columns Formulation, DrugName, SubCat and SubCat1.

```
UPDATE clean_projectfinaldata
SET
  Formulation = CASE WHEN Formulation = '' THEN 'unknown' ELSE Formulation END;


UPDATE clean_projectfinaldata
SET
  DrugName = CASE WHEN DrugName = '' THEN 'unknown' ELSE DrugName END;


UPDATE clean_projectfinaldata
SET
  SubCat = CASE WHEN SubCat = '' THEN 'unknown' ELSE SubCat END;


UPDATE clean_projectfinaldata
SET
  SubCat1 = CASE WHEN SubCat1 = '' THEN 'unknown' ELSE SubCat1 END;
```

Showing the columns after replacing the missing values with ‘unknown’:

```
SELECT Formulation, DrugName, SubCat, SubCat1 FROM clean_projectfinaldata;
```

Formulation	DrugName	SubCat	SubCat1
Form1	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Form1	unknown	unknown	unknown
Form1	unknown	unknown	unknown
Form2	SODIUM CHLORIDE 0.9%	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Form1	unknown	unknown	unknown
Form1	PARACETAMOL 1GM IV INJ	INJECTIONS	CENTRAL NERVOUS SYSTEM
unknown	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Form1	SODIUM CHLORIDE IVF 100ML	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Form1	unknown	unknown	unknown
Form1	SODIUM CHLORIDE 0.9%	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Form1	PARACETAMOL 1GM IV INJ	INJECTIONS	CENTRAL NERVOUS SYSTEM
Form1	LIGNOCAINE HYDROCHLORIDE 2% INJ	OINTMENTS, CREAMS & GELS	ANAESTHETICS
Form1	POLYANTIBIOTIC RESISTANT BACILLUS ...	SYRUP & SUSPENSION	GASTROINTESTINAL & HEPATOBILIARY SYST...
Form1	HUMAN ALBUMIN 25% INJ	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
unknown	POTASSIUM CHLORIDE 150MG	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTIONS
Form1	SEVOFLURANE 99.97%	LIQUIDS & SOLUTIONS	ANAESTHETICS

9. Creating a new table called `missing_values` by selecting rows from `clean_projectfinaldata` where any of the columns (`Formulation`, `DrugName`, `SubCat`, or `SubCat1`) has the value 'unknown'.

```
CREATE TABLE missing_values AS

SELECT *

FROM clean_projectfinaldata

WHERE Formulation = 'unknown'

  OR DrugName = 'unknown'

  OR SubCat = 'unknown'

  OR SubCat1 = 'unknown';
```

Showing missing_values table and count of records with at least one or more missing values:

```
SELECT * FROM missing_values;

SELECT COUNT(*) AS missing_records_count FROM missing_values;
```

Output:

Typeofsales	Patient_ID	Specialisation	Dept	Dateofbill	Quantity	ReturnQuantity	Final_Cost	Final_Sales	RtnMRP	Formulation	DrugName	SubCat	SubCat1
Sale	12018111286	Specialisation25	Department1	2022-09-07	1	0	49.352	60.8	0	Form1	unknown	unknown	unknown
Sale	12018097033	Specialisation20	Department1	2022-09-17	2	0	40.34	81.1	0	Form1	unknown	unknown	unknown
Sale	12018106746	Specialisation20	Department1	2022-10-14	2	0	40.34	81.1	0	Form1	unknown	unknown	unknown
Return	12018109493	Specialisation5	Department1	2022-09-17	0	1	64.864	0	97	unknown	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018116917	Specialisation25	Department1	2022-11-07	1	0	49.956	62.8	0	Form1	unknown	unknown	unknown
Sale	12018109292	Specialisation5	Department1	2022-09-16	4	0	52.544	181.12	0	unknown	POTASSIUM CHLORIDE 150MG	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTIONS
Return	12018072512	Specialisation20	Department1	2022-11-07	0	1	57.408	0	62	unknown	CALCIUM 250MG + VITAMIN D3 125IU	TABLETS & CAPSULES	VITAMINS & MINERALS
Sale	12018084566	Specialisation4	Department2	2022-01-31	4	0	77.408	243.2	0	Form1	unknown	unknown	unknown
Sale	12018086686	Specialisation4	Department1	2022-08-04	1	0	49.352	60.8	0	Form1	unknown	unknown	unknown
Sale	12018118093	Specialisation11	Department1	2022-11-04	4	0	79.828	251.2	0	Form1	unknown	unknown	unknown

missing_records_count
2181

Observation:

A total of 2181 records with missing values have been retrieved. These records need to be rechecked with the client for further discussion. We can refer to the 'missing_values' table for reviewing the specific data.

10. Identifying duplicate rows based on Patient_ID, Dateofbill, and DrugName excluding rows where the DrugName is 'unknown'.

```
SELECT Patient_ID, Dateofbill, DrugName, COUNT(*)

FROM clean_projectfinaldata

WHERE DrugName <> 'unknown'

GROUP BY Patient_ID, Dateofbill, DrugName

HAVING COUNT(*) > 1;
```

Output:

Patient_ID	Dateofbill	DrugName	COUNT(*)
12018044140	2022-07-30	MEROPENEM 1GM INJ	2
12018123919	2022-12-23	FUROSEMIDE 10MG/ML	2
12018120229	2022-11-22	PHENERAMINE MALEATE 22.75MG/2ML	2
12018073408	2022-02-03	MEROPENEM 1GM INJ	2
12018101171	2022-08-04	PIPERACILLIN 1GM + TAZOBACTAM 125MG	2
12018082789	2022-03-19	ONDANSETRON 2MG/ML	2
12018085615	2022-05-18	LEVOSALBUTAMOL 50MCG + IPRATROPIUM BROMIDE 20MCG INHALER	2
12018095828	2022-05-05	HUMAN GAMMA GLOBULIN 100ML INJ	2
12018080259	2022-03-07	PANTOPRAZOLE 40MG INJ	2
12018100335	2022-07-07	MULTIPLE ELECTROLYTES 500ML IVF	2
12018053294	2022-02-17	SODIUM CHLORIDE 0.45%	2
12018074192	2022-05-18	MORPHINE SULPHATE 10MG/ML INJ	2
12018091513	2022-04-19	CEFAZOLIN 500MG	2
12018102578	2022-12-14	POTASSIUM CHLORIDE 150MG	2
12018088971	2022-03-04	ONDANSETRON 2MG/ML	2
12018072438	2022-12-12	HYDROCORTISONE 100MG INJ	2
12018075690	2022-03-05	NUTRITIONAL SUPPLEMENT POWDER	2
12018084814	2022-01-29	SODIUM CHLORIDE 0.9%	2

11. Removing the duplicate rows from clean_projectfinaldata table and counting the remaining rows.

```
DELETE FROM clean_projectfinaldata
WHERE (Patient_ID, Dateofbill, DrugName) IN (
   SELECT t.Patient_ID, t.Dateofbill, t.DrugName
   FROM (
      SELECT Patient_ID, Dateofbill, DrugName
      FROM clean_projectfinaldata
      GROUP BY Patient_ID, Dateofbill, DrugName
      HAVING COUNT(*) > 1
   ) AS t
);
SELECT COUNT(*) AS total_rows FROM clean_projectfinaldata;
```

Output:

total_rows
13967

Observation:

After removing the duplicate rows from the `clean_projectfinaldata` table, the total number of rows has been reduced from 14,218 to 13,967.

12. In a normal distribution, approximately 68%, 95%, and 99.7% of the data falls within one, two, and three standard deviations of the mean respectively. By using three standard deviations as the threshold for removing outliers, we are effectively removing data points that are more than three standard deviations away from the mean.

```
CREATE TABLE new_table AS
SELECT *
FROM clean_projectfinaldata
WHERE Quantity BETWEEN
  (SELECT AVG(Quantity) - 3 * STDDEV(Quantity) FROM clean_projectfinaldata)
  AND
  (SELECT AVG(Quantity) + 3 * STDDEV(Quantity) FROM clean_projectfinaldata)
  AND ReturnQuantity BETWEEN
  (SELECT AVG(ReturnQuantity) - 3 * STDDEV(ReturnQuantity) FROM clean_projectfinaldata)
  AND
  (SELECT AVG(ReturnQuantity) + 3 * STDDEV(ReturnQuantity) FROM clean_projectfinaldata)
  AND Final_Cost BETWEEN
  (SELECT AVG(Final_Cost) - 3 * STDDEV(Final_Cost) FROM clean_projectfinaldata)
  AND
  (SELECT AVG(Final_Cost) + 3 * STDDEV(Final_Cost) FROM clean_projectfinaldata)
  AND Final_Sales BETWEEN
  (SELECT AVG(Final_Sales) - 3 * STDDEV(Final_Sales) FROM clean_projectfinaldata)
  AND
  (SELECT AVG(Final_Sales) + 3 * STDDEV(Final_Sales) FROM clean_projectfinaldata)
  AND RtnMRP BETWEEN
  (SELECT AVG(RtnMRP) - 3 * STDDEV(RtnMRP) FROM clean_projectfinaldata)
  AND
  (SELECT AVG(RtnMRP) + 3 * STDDEV(RtnMRP) FROM clean_projectfinaldata);
```

Showing the count of rows after removing the outliers:

```
SELECT COUNT(*) AS total_rows FROM new_table;
```

Output:

total_rows
13399

Observation:

Outliers have been excluded from the ‘clean_projectfinaldata’ table and the filtered records have been stored in the ‘new_table’. The new_table’ now contains 13,399 records, excluding the outliers. To save the cleaned dataset with 13,399 rows in CSV format, we can go to "Query" -> "Export Results" -> "Save as CSV". We can specify the file name and location to save the CSV file, allowing us to import it for future exploratory data analysis (EDA) or other purposes.