SELECT \* FROM Cosmetic

SELECT \* FROM Company

SELECT \* FROM Brand

--DATATYPE FOR EACH COLUMN

SHOW COLUMNS IN TABLE COSMETICCHEMICAL.PUBLIC.COSMETIC;

--Since Primary Key is combined from CDPHId,CSFId,SubCategoryId,ChemicalId

--33973 NULL VALUES FOR CSFID, so I need to process before generate CosmeticId

SELECT COUNT(\*) FROM COSMETIC

WHERE CSFID IS NULL;

--FILL-IN NULL VALUES WITH 0

UPDATE COSMETIC

SET CSFID=0

WHERE CSFID IS NULL;

--FILL-IN NULL VALUES WITH N/A

--#33973 IS UPDATED

UPDATE COSMETIC

SET CSF='N/A'

WHERE CSF IS NULL;

--Add column CosmeticId (Primary key, combined from CDPHId,CSFId,SubCategoryId,ChemicalId)

ALTER TABLE COSMETIC

ADD COLUMN COSMETICID VARCHAR(255);

--SET AS PRIMARY KEY

ALTER TABLE COSMETIC

ADD CONSTRAINT PK\_COSMETICID PRIMARY KEY (COSMETICID);

--114635 DATA IS UPDATED

UPDATE COSMETIC

SET COSMETICID=CONCAT(CDPHId, '-', CSFId, '-', SubCategoryId, '-', ChemicalId)

--COSMETICID IN THE END of table

SELECT \* FROM COSMETIC

--114377 distinct CosmeticId

SELECT DISTINCT COSMETICID FROM COSMETIC

SELECT \* FROM COSMETIC

--CREATE BRANDID,and Create Brand Table-------------------------------------------------------

-- Add BrandID

ALTER TABLE Cosmetic

ADD BrandID INT;

-- Create BrandID column with unique identifiers

UPDATE Cosmetic c --Using Subquery

SET c.BrandID = rb.BrandID

FROM (

SELECT

ROW\_NUMBER() OVER (ORDER BY BrandName) AS BrandID,

BrandName

FROM

Cosmetic

GROUP BY

BrandName

) AS rb

WHERE c.BrandName = rb.BrandName;

-- Create Brand table

CREATE TABLE Brand AS

SELECT

BRANDID,

BRANDNAME

FROM

COSMETIC

GROUP BY

BRANDID,

BRANDNAME

ORDER BY

BRANDID;

--delete null values

DELETE FROM

BRAND

WHERE

BRANDID IS NULL

-- SORT AND UPDATE TABLE

CREATE TABLE BRAND\_sorted AS

SELECT \*

FROM BRAND

ORDER BY BrandId ASC;

DROP TABLE BRAND;

ALTER TABLE BRAND\_sorted RENAME TO BRAND;

--FILL NULL BRANDNAME

UPDATE COSMETIC COS

SET COS.BRANDNAME=B.BRANDNAME

FROM BRAND B

WHERE COS.BRANDID=B.BRANDID

--CHECK RESULT

SELECT \* FROM BRAND

--FILL-IN NULL VALUES OF BRANDNAME AND BRANDID IN COSMETIC TABLES

--177 NULL VALUES UPDATED.

UPDATE COSMETIC

SET BRANDNAME='N/A'

WHERE BRANDNAME IS NULL;

UPDATE COSMETIC

SET BRANDID=0

WHERE BRANDID IS NULL;

---------------------------------------------------------------------------------------------------------

--There are no Null Values in Column Primarycategory and Subcategory, and there's relationship between

--so we create the foreign key for subcategoryid

SELECT \* FROM COSMETIC

WHERE SUBCATEGORY IS NULL;

SELECT \* FROM COSMETIC

WHERE PRIMARYCATEGORY IS NULL

--Create SUBCATEGORY table-------------------------------------------------------------------------

CREATE TABLE SUB\_CATEG(

SUBCATEGORYID INT PRIMARY KEY,

SUBCATEGORY VARCHAR(255)

);

INSERT INTO SUB\_CATEG (

SUBCATEGORYID,

SUBCATEGORY

)

SELECT DISTINCT

SUBCATEGORYID,

SUBCATEGORY

FROM

COSMETIC

ORDER BY SUBCATEGORYID;

--CHECK RESULT

SELECT \* FROM SUB\_CATEG

--Create PRIMARYCATEGORY table----------------------------------------------------------------------------

CREATE TABLE PRIMARY\_CATEG (

PRIMARYCATEGORYID INT PRIMARY KEY,

PRIMARYCATEGORY VARCHAR(255),

SUBCATEGORYID INT,

FOREIGN KEY (SUBCATEGORYID) REFERENCES SUB\_CATEG(SUBCATEGORYID)

);

INSERT INTO PRIMARY\_CATEG (PRIMARYCATEGORYID,PRIMARYCATEGORY,SUBCATEGORYID)

SELECT DISTINCT PRIMARYCATEGORYID,PRIMARYCATEGORY,SUBCATEGORYID

FROM

COSMETIC

ORDER BY

PRIMARYCATEGORYID

--CHECK RESULT

SELECT \* FROM PRIMARY\_CATEG

---Create Product Table---------------------------------------------------------------------

--Since primarycateg table is created, we can create product table with FK for PrimarycategoryId

--FOUND OUT EACH PRODUCT HAS MULTIPLE PRIMARYCATEGORY.

SELECT

PRODUCTID,

PRODUCTNAME,

LISTAGG(DISTINCT PRIMARYCATEGORYID, ', ') WITHIN GROUP (ORDER BY PRIMARYCATEGORYID) AS PRIMARYCATEGORYIDS

FROM COSMETIC

GROUP BY PRODUCTID, PRODUCTNAME;

--CREATE PRODUCT TABLE

CREATE TABLE PRODUCT(

CDPHID INT PRIMARY KEY,

PRODUCTNAME VARCHAR(255),

PRIMARYCATEGORYID INT,

FOREIGN KEY (PRIMARYCATEGORYID) REFERENCES PRIMARY\_CATEG(PRIMARYCATEGORYID)

) ;

INSERT INTO PRODUCT(CDPHID,PRODUCTNAME,PRIMARYCATEGORYID)

SELECT DISTINCT CDPHID,PRODUCTNAME,PRIMARYCATEGORYID

FROM COSMETIC

ORDER BY CDPHID;

--CHECK RESULT

SELECT \* FROM PRODUCT

-- Create Company table-----------------------------------------------------------------------------

--FOUND OUT ONLY WHEN COMPANYID=91 OR COMPANYID=717 HAS BRANDID IS NULL

SELECT DISTINCT COMPANYID,BRANDID

FROM COSMETIC

WHERE BRANDID=0

--Each Company has several BrandName

--Create company table

CREATE TABLE COMPANY AS

SELECT DISTINCT

COMPANYID,

COMPANYNAME,

BRANDID

FROM

COSMETIC

GROUP BY

COMPANYID, COMPANYNAME,BRANDID

ORDER BY

COMPANYID ASC

--CHECK RESULT

SELECT \* FROM COMPANY

--CREATE TABLE ColorScentFlavor-------------------------------------------

CREATE TABLE ColorScentFlavor AS

SELECT DISTINCT

CSFID,

CSF

FROM

COSMETIC

WHERE

CSFID IS NOT NULL

AND CSF IS NOT NULL

ORDER BY CSFID ASC

--CHECK RESULT

SELECT \* FROM ColorScentFlavor;

--CREATE CAS TABLE---------------------------------------------------------------------------------

--#6476 NULL VALUES IN CASNUMBER COLUMN

SELECT CASNUMBER FROM COSMETIC

WHERE CASNUMBER IS NULL;

--CREATE CAS TABLE

CREATE TABLE CAS (

CASID INT PRIMARY KEY,

CASNUMBER VARCHAR(255)

);

--127 VALUES UPDATED

INSERT INTO CAS(CASID,CASNUMBER)

SELECT DISTINCT

CASID,CASNUMBER

FROM

COSMETIC

WHERE CASNUMBER IS NOT NULL;

--CHECK RESULT

SELECT \* FROM CAS;

--Create TIME-related table-------------------------------------------------------------------------

--CreateMOSTRECENTDATEREPORTED Table---------------------------------

ALTER TABLE COSMETIC

ADD COLUMN MOSTRECENTDATEREPORTEDID INT ;

CREATE TABLE Most\_Recent\_Date\_Reported(

MOSTRECENTDATEREPORTEDID INT PRIMARY KEY,

YEAR INT,

MONTH INT,

DAY INT

)

--Create a temporary table for generate unique mostrecentdatereported\_Id

CREATE TEMPORARY TABLE TempRankedDates AS

SELECT ROW\_NUMBER() OVER (ORDER BY MostRecentDateReported) AS SID,

MostRecentDateReported

FROM (

SELECT DISTINCT MostRecentDateReported

FROM COSMETIC

);

UPDATE COSMETIC

SET COSMETIC.MOSTRECENTDATEREPORTEDID = T.SID

FROM TempRankedDates T

WHERE COSMETIC.MostRecentDateReported = T.MostRecentDateReported;

--Insert into table MRDR

INSERT INTO Most\_Recent\_Date\_Reported (MOSTRECENTDATEREPORTEDID,YEAR,MONTH,DAY)

SELECT DISTINCT

MOSTRECENTDATEREPORTEDID,

YEAR(MostRecentDateReported) AS YEAR,

MONTH(MOSTRECENTDATEREPORTED) AS MONTH,

DAY(MOSTRECENTDATEREPORTED) AS DAY

FROM COSMETIC

ORDER BY MOSTRECENTDATEREPORTEDID;

SELECT \*

FROM MOST\_RECENT\_DATE\_REPORTED;

DROP TABLE TEMPRANKEDDATES;

--Create InitialDateReport Table---------------------------------

ALTER TABLE COSMETIC

ADD COLUMN InitialDateReportedId INT ;

CREATE TABLE Initial\_Date\_Report(

InitialDateReportedId INT PRIMARY KEY,

YEAR INT,

MONTH INT,

DAY INT

)

--CHECK NULL VALUES,NO NULL VALUES

SELECT initialdatereported

FROM COSMETIC

WHERE initialdatereported IS NULL;

--CREATE TEMPORARY TABLE

--Using subquery

CREATE TEMPORARY TABLE TempRankedDates AS

SELECT ROW\_NUMBER() OVER (ORDER BY InitialDateReported) AS SID,

InitialDateReported

FROM (

SELECT DISTINCT InitialDateReported

FROM COSMETIC

);

UPDATE COSMETIC

SET COSMETIC.InitialDateReportedId = T.SID

FROM TempRankedDates T

WHERE COSMETIC.InitialDateReported = T.InitialDateReported;

--INSERT VALUES INTO TABLE

INSERT INTO Initial\_Date\_Report (InitialDateReportedId,YEAR,MONTH,DAY)

SELECT DISTINCT

InitialDateReportedId ,

YEAR(InitialDateReported) AS YEAR,

MONTH(InitialDateReported) AS MONTH,

DAY(InitialDateReported) AS DAY

FROM COSMETIC

ORDER BY InitialDateReportedId ;

--CHECK RESULT

SELECT \* FROM INITIAL\_DATE\_REPORT;

DROP TABLE TEMPRANKEDDATES;

--Create ChemicalDateRemoved Table---------------------------------

ALTER TABLE COSMETIC

ADD COLUMN ChemicalDateRemovedId INT ;

CREATE TABLE Chemical\_Date\_Removed(

ChemicalDateRemovedId INT PRIMARY KEY,

YEAR INT,

MONTH INT,

DAY INT

);

INSERT INTO CHEMICAL\_DATE\_REMOVED (CHEMICALDATEREMOVEDID,YEAR,MONTH,DAY)

VALUES(0,0,0,0)

--CHECK NULL VALUES,#2985 NULL VALUES

SELECT \* FROM COSMETIC

WHERE ChemicalDateRemoved IS NOT NULL

--FOUND MANUAL FILL-IN MISTAKE WHERE The ChemicalDateRemoved is 2103-12-05,2104-05-02

UPDATE COSMETIC

SET CHEMICALDATEREMOVED='2013-12-05'

WHERE CHEMICALDATEREMOVED='2103-12-05';

UPDATE COSMETIC

SET CHEMICALDATEREMOVED='2014-05-02'

WHERE CHEMICALDATEREMOVED='2104-05-02';

--CREATE TEMPORARY TABLE

CREATE TEMPORARY TABLE TempRankedDates AS

SELECT ROW\_NUMBER() OVER (ORDER BY ChemicalDateRemoved) AS SID,

ChemicalDateRemoved

FROM (

SELECT DISTINCT ChemicalDateRemoved

FROM COSMETIC

WHERE CHEMICALDATEREMOVED IS NOT NULL

);

-- UPDATE COSMETIC ChemicalDateRemovedId

UPDATE COSMETIC

SET COSMETIC.ChemicalDateRemovedId = T.SID

FROM

TempRankedDates T

WHERE

COSMETIC.ChemicalDateRemoved = T.ChemicalDateRemoved

AND

COSMETIC.ChemicalDateRemoved IS NOT NULL;

--INSERT VALUES INTO TABLE

INSERT INTO Chemical\_Date\_Removed (ChemicalDateRemovedId,YEAR,MONTH,DAY)

SELECT DISTINCT

ChemicalDateRemovedId ,

YEAR(ChemicalDateRemoved) AS YEAR,

MONTH(ChemicalDateRemoved) AS MONTH,

DAY(ChemicalDateRemoved) AS DAY

FROM COSMETIC

WHERE CHEMICALDATEREMOVED IS NOT NULL

ORDER BY ChemicalDateRemovedId;

--CHECK RESULT

SELECT \* FROM CHEMICAL\_DATE\_REMOVED

--SET ALL NULL VALUES ID TO 0

UPDATE COSMETIC

SET CHEMICALDATEREMOVEDID=0

WHERE CHEMICALDATEREMOVED IS NULL;

--CHECK RESULT OF SETTING 0 TO NULL VALUE

SELECT\* FROM COSMETIC;

DROP TABLE TEMPRANKEDDATES;

--Create ChemicalUpdatedAt Table---------------------------------

--NO NULL VALUE

SELECT \*

FROM COSMETIC

WHERE ChemicalUpdatedAt IS NULL;

--ADD COLUMN TO ORIGIN DATA

ALTER TABLE COSMETIC

ADD COLUMN ChemicalUpdatedAtID INT ;

--CREATE TABLE

CREATE TABLE Chemical\_Updated\_At(

ChemicalUpdatedAtId INT PRIMARY KEY,

YEAR INT,

MONTH INT,

DAY INT

);

--CREATE TEMPORARY TABLE

CREATE TEMPORARY TABLE TempRankedDates AS

SELECT ROW\_NUMBER() OVER (ORDER BY ChemicalUpdatedAt) AS SID,

ChemicalUpdatedAt

FROM (

SELECT DISTINCT ChemicalUpdatedAt

FROM COSMETIC

);

UPDATE COSMETIC

SET COSMETIC. ChemicalUpdatedAtId = T.SID

FROM TempRankedDates T

WHERE COSMETIC. ChemicalUpdatedAt= T.ChemicalUpdatedAt;

--INSERT VALUES INTO TABLE

INSERT INTO Chemical\_Updated\_At (ChemicalUpdatedAtId,YEAR,MONTH,DAY)

SELECT DISTINCT

ChemicalUpdatedAtId ,

YEAR(ChemicalUpdatedAt) AS YEAR,

MONTH(ChemicalUpdatedAt) AS MONTH,

DAY(ChemicalUpdatedAt) AS DAY

FROM COSMETIC

ORDER BY ChemicalUpdatedAtId;

--CHECK RESULT

SELECT \* FROM Chemical\_Updated\_At;

DROP TABLE TEMPRANKEDDATES;

--Create ChemicalCreatedAt Table---------------------------------

--No NULL VALUES

SELECT \*

FROM COSMETIC

WHERE ChemicalCreatedAt IS NULL;

--ADD COLUMN TO ORIGIN DATA

ALTER TABLE COSMETIC

ADD COLUMN ChemicalCreatedAtID INT ;

--CREATE TABLE

CREATE TABLE Chemical\_Created\_At(

ChemicalCreatedAtID INT PRIMARY KEY,

YEAR INT,

MONTH INT,

DAY INT

);

--CREATE TEMPORARY TABLE

CREATE TEMPORARY TABLE TempRankedDates AS

SELECT ROW\_NUMBER() OVER (ORDER BY ChemicalCreatedAt) AS SID,

ChemicalCreatedAt

FROM (

SELECT DISTINCT ChemicalCreatedAt

FROM COSMETIC

);

UPDATE COSMETIC

SET COSMETIC.ChemicalCreatedAtId = T.SID

FROM TempRankedDates T

WHERE COSMETIC.ChemicalCreatedAt= T.ChemicalCreatedAt;

--INSERT VALUES INTO TABLE

INSERT INTO Chemical\_Created\_At (ChemicalCreatedAtId,YEAR,MONTH,DAY)

SELECT DISTINCT

ChemicalCreatedAtId ,

YEAR(ChemicalCreatedAt) AS YEAR,

MONTH(ChemicalCreatedAt) AS MONTH,

DAY(ChemicalCreatedAt) AS DAY

FROM COSMETIC

ORDER BY ChemicalCreatedAtId;

--CHECK RESULT

SELECT \* FROM Chemical\_Created\_At;

DROP TABLE TEMPRANKEDDATES;

--Create DiscontinuedDate Table---------------------------------

--HAS NULL VALUES

SELECT \*

FROM COSMETIC

WHERE DiscontinuedDate IS NOT NULL; --#12920 NOT NULL

--ADD COLUMN TO ORIGIN DATA

ALTER TABLE COSMETIC

ADD COLUMN DiscontinuedDateId INT ;

--CREATE TABLE

CREATE TABLE Discontinued\_Date(

DiscontinuedDateId INT PRIMARY KEY,

YEAR INT,

MONTH INT,

DAY INT

);

INSERT INTO Discontinued\_Date (DiscontinuedDateID,YEAR,MONTH,DAY)

VALUES(0,0,0,0);

--CREATE TEMPORARY TABLE

CREATE TEMPORARY TABLE TempRankedDates AS

SELECT ROW\_NUMBER() OVER (ORDER BY DiscontinuedDate) AS SID,

DiscontinuedDate

FROM (

SELECT DISTINCT DiscontinuedDate

FROM COSMETIC

WHERE DiscontinuedDate IS NOT NULL

);

-- UPDATE COSMETIC ChemicalDateRemovedId

UPDATE COSMETIC

SET COSMETIC.DiscontinuedDateId = T.SID

FROM

TempRankedDates T

WHERE

COSMETIC.DiscontinuedDate = T.DiscontinuedDate

AND

COSMETIC.DiscontinuedDate IS NOT NULL;

--INSERT VALUES INTO TABLE

INSERT INTO Discontinued\_Date (DiscontinuedDateId,YEAR,MONTH,DAY)

SELECT DISTINCT

DiscontinuedDateId ,

YEAR(DiscontinuedDate) AS YEAR,

MONTH(DiscontinuedDate) AS MONTH,

DAY(DiscontinuedDate) AS DAY

FROM COSMETIC

WHERE DiscontinuedDate IS NOT NULL

ORDER BY DiscontinuedDateId;

--CHECK RESULT

SELECT \* FROM Discontinued\_Date;

--SET ALL NULL VALUES ID TO 0

UPDATE COSMETIC

SET DiscontinuedDateID=0

WHERE DiscontinuedDate IS NULL;

--CHECK RESULT OF SETTING 0 TO NULL VALUE

SELECT\* FROM COSMETIC;

DROP TABLE TEMPRANKEDDATES

-------------FINISHED ALL TABLES SPLIT----------------------------------------

-------------NOW FINAL CLEAN THE ORGINAL DATA--------------------------------------

SELECT \* FROM COSMETIC;

--CREATE FINAL CLEANED TABLE AND REMAIN THE ORIGINAL ONE 'COSMETIC' TABLE IN CASE TO USE.

CREATE TABLE COSMETIC\_CLEANED AS

SELECT

COSMETICID,

CDPHId,

COMPANYID,

CSFID,

CASID,

CHEMICALID,

InitialDateReportedId,

MostRecentDateReportedId,

DiscontinuedDateId,

ChemicalCreatedAtId,

ChemicalUpdatedAtId,

ChemicalDateRemovedId,

CHEMICALNAME,

ChemicalCount,

FROM

COSMETIC;

--CHECK RESULT

SELECT \* FROM COSMETIC\_CLEANED;

-----------------------Business Questions----------------------------------

--Business Question1:CHEMICAL RANKS

--year=2009 is index<= 116

select \* from initial\_date\_report

where year=2009

--CHECK INDEX FOR YEAR 2009~2O20

--2009 1~116

--2010 117~328

--2011 329~501

--2012 502~666

--2013 667~863

--2014 864~1075

--2015 1076~1307

--2016 1308~1493

--2017 1494~1708

--2018 1709~1932

--2019 1933~2177

--2020 2178~2274

SELECT YEAR,MIN(INITIALDATEREPORTEDID),MAX(INITIALDATEREPORTEDID)

FROM INITIAL\_DATE\_REPORT

GROUP BY YEAR

ORDER BY YEAR ASC;

--check company chemical count and rank.

SELECT COM.COMPANYNAME,SUM(C.CHEMICALCOUNT)

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY COM ON COM.COMPANYID=C.COMPANYID

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--CHEMICAL RANKS

SELECT IDP.YEAR AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS CHEMICAL\_AMOUNTS

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

GROUP BY IDP.YEAR,COM.COMPANYNAME

ORDER BY IDP.YEAR ASC

--2009 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=116

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2010 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=328 AND IDP.InitialDateReportedId>=117

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2011 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=501 AND IDP.InitialDateReportedId>=329

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2012 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=666 AND IDP.InitialDateReportedId>=502

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2013 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=863 AND IDP.InitialDateReportedId>=667

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2014 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=1075 AND IDP.InitialDateReportedId>=864

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2015 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=1307 AND IDP.InitialDateReportedId>=1076

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2016 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=1493 AND IDP.InitialDateReportedId>=1308

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2017 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=1708 AND IDP.InitialDateReportedId>=1494

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2018 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=1932 AND IDP.InitialDateReportedId>=1709

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2019 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=2177 AND IDP.InitialDateReportedId>=1933

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--2020 CHEMICAL RANKS

SELECT MAX(IDP.YEAR)AS YEAR,COM.COMPANYNAME,SUM(C.CHEMICALCOUNT) AS Chemical\_Counts

FROM COSMETIC\_CLEANED C

LEFT JOIN

COMPANY AS COM ON C.COMPANYID=COM.COMPANYID

LEFT JOIN

INITIAL\_DATE\_REPORT IDP ON IDP.INITIALDATEREPORTEDID=C.INITIALDATEREPORTEDID

WHERE IDP.InitialDateReportedId<=2274 AND IDP.InitialDateReportedId>=2178

GROUP BY COM.COMPANYNAME

ORDER BY SUM(C.CHEMICALCOUNT) DESC;

--Business Question2

--Top variations in product ?

SELECT

PRO.PRODUCTNAME,

LISTAGG(DISTINCT CSL.CSFID, '// ') WITHIN GROUP (ORDER BY CSL.CSFID) AS CSFIDs,

COUNT(DISTINCT CSL.CSFID)

FROM

COLORSCENTFLAVOR CSL

LEFT JOIN

COSMETIC\_CLEANED AS C ON CSL.CSFID = C.CSFID

LEFT JOIN

PRODUCT AS PRO ON PRO.CDPHID = C.cdphid

GROUP BY

PRO.PRODUCTNAME

ORDER BY COUNT(DISTINCT CSL.CSFID) DESC

LIMIT 10;

--Business Question3

SELECT \* FROM