PROJECT - 2

(BUAN 6320: Database Foundations for Business Analytics)

Topic: Retail Data of Turkish Sector

Abhishek Arya	axa220149@utdallas.edu
Manan Mistry	mxm220108@utdallas.edu
Tanishka Patel	txp210040@utdallas.edu
Het Shah	hrs220002@utdallas.edu
Shekhar Vashist	sxv220008@utdallas.edu

Project made under the guidance of: Farzad Kamalzadeh

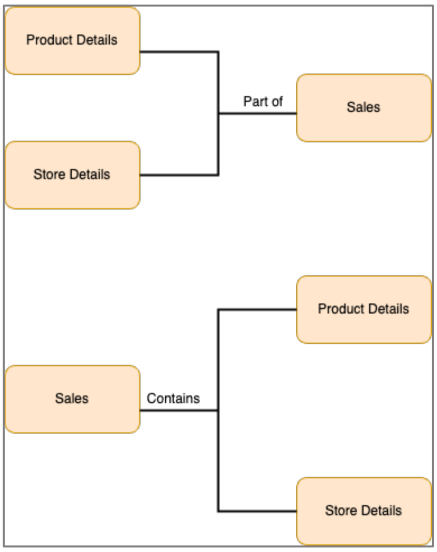
SECTION 1: DESIGN A DATABASE

Queries 1 and 2 are completed in project 1.

3. SCHEMA DESIGN

a. Find entities, their attributes, their primary keys, and relationships between them.

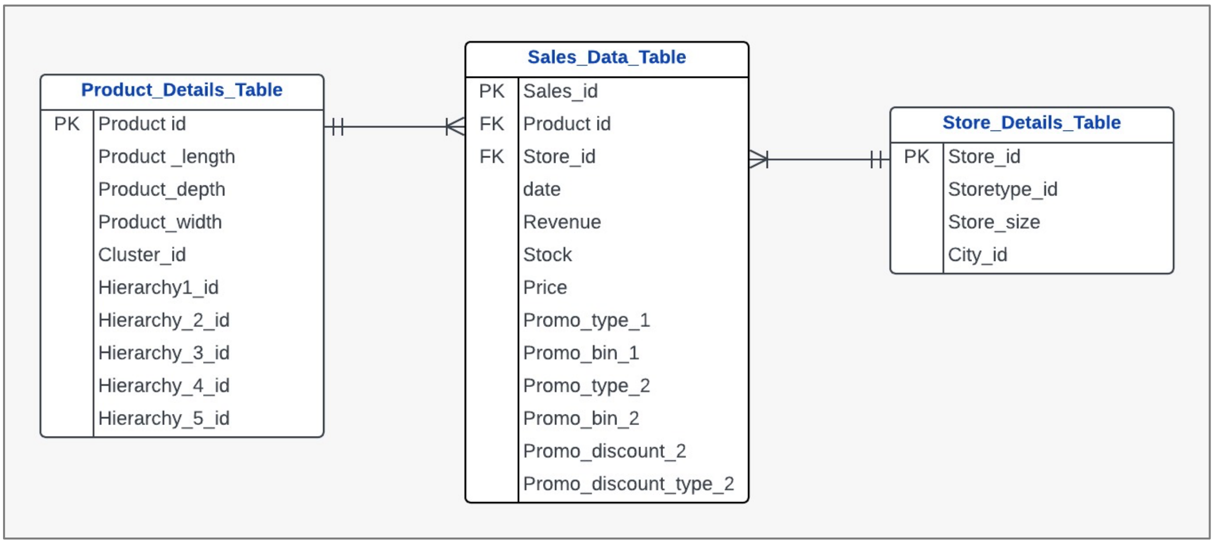
ENTITIES	ATTRIBUTES	PRIMARY KEYS
Product_Details_Table	Product_id Product_length Product_width Cluster_id Hierarchy1_id Hierarchy2_id Hierarchy3_id Hierarchy4_id Hierarchy5_id	Product_id
Store_Details_Table	Store_id Storetype_id Store_size City_id	Store_id
Sales_Data_Table	Sales_id Store_id Product_id Date Sales Revenue Stock Price Promo_type_1 Promo_bin_1 Promo_type_2 Promo_bin_2	Sales_id



b. Model all the constraints you believe should be there in your schema.

1. Each Sales record must contain only one Product Details Table record and one Store Details Table record.
2. Each Store Details Table can be part of 1 or more Sales records
3. Each Product Details Table can be part of 1 or more Sales records.

c. Draw and ER diagram of your dataset.



Product_Details_Table

PK	Product id
	Product _length
	Product_depth
	Product_width
	Cluster_id
	Hierarchy1_id
	Hierarchy_2_id
	Hierarchy_3_id
	Hierarchy_4_id
	Hierarchy_5_id

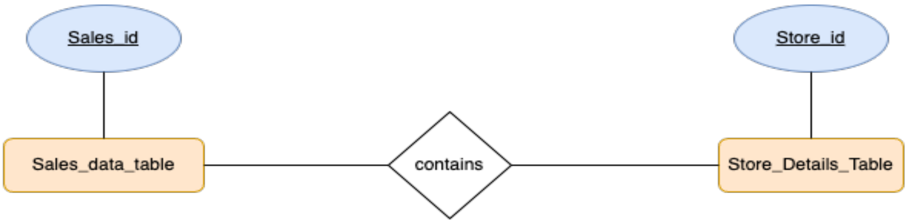
Sales_Data_Table

PK	Sales_id
FK	Product id
FK	Store_id
	date
	Revenue
	Stock
	Price
	Promo_type_1
	Promo_bin_1
	Promo_type_2
	Promo_bin_2
	Promo_discount_2
	Promo_discount_type_2

Store_Details_Table

PK	Store_id
	Storetype_id
	Store_size
	City_id

d. Translate your ER diagram into relations.



Sales Data

(sales_id, product_id,
store_id,
date,
sales,
revenue,
stock,
price,
promotype_1,
promobin_1,
promotype_2,
promobin_2)

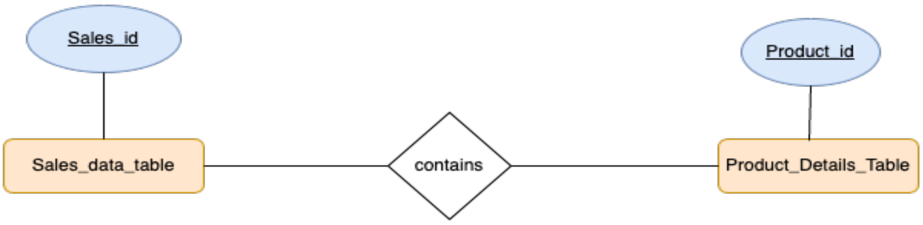
Contains

sales_id ,

store_id

Store Details

(Store_id,
storetype_id,
store_size,
city_id)



Sales Data

(sales_id, product_id,
store_id,
date,
sales,
revenue,
stock,
price,
promotype_1,
promobin_1,
promotype_2,
promobin_2)

Contains

sales_id ,

product_id

Product Details

(product_id,
product_length,
product_depth,
product_width,
cluster_id,
hierarchy1_id,
hierarchy2_id,
hierarchy3_id,
hierarchy4_id,
hierarchy5_id)

4. SCHEMA NORMALIZATION

a. Functional Dependencies:

Sales Table

sales_id → product_id, store_id, date

product_id, store_id, date → sales_id (product id and store id gives a unique sales id)

sales_id → product_id, store_id, date, sales, revenue, stock, price, promo_type_1, promo_bin_1, promo_type_2

Product Table

hierarchy5_id → hierarchy4_id

hierarchy4_id → hierarchy3_id

hierarchy3_id → hierarchy2_id

hierarchy2_id → hierarchy1_id

product_id → product_length, product_width, product_depth, cluster_id,

hierarchy1_id, hierarchy2_id, hierarchy3_id, hierarchy4_id, hierarchy5_id

hierarchy5_id → hierarchy4_id, hierarchy3_id, hierarchy2_id, hierarchy1_id

Store Details

store_id → storetype_id, store_size, city_id

b. Minimal Keys:

Currently, sales_id aside, the primary key for sales {product_id, store_id, date} and that is minimal because subsets of this are not candidate keys. The same goes for product_id is the primary key for product details and store_id is the primary key for store details.

Each minimal key is a unique combination of attributes that can identify a tuple in a relation without any redundancy. Identifying minimal keys is essential for enforcing data integrity and avoiding redundancy in the database schema.

c. Check if your schema is in BCNF (Boyce-Codd Normal Form)

A relation is in BCNF if, for every non-trivial functional dependency X -> Y, X is a super key. Before checking if the schema is in BCNF (Boyce-Codd Normal Form), let us recap the functional dependencies which we identified for each table: The functional dependencies indicate that the schema is in BCNF, as the left side of each functional dependency represents the primary key of each entity.

Sales Table

sales_id → product_id, store_id, date

product_id, store_id, date → sales_id

sales_id → product_id, store_id, date, sales, revenue, stock, price, promo_type_1, promo_bin_1, promo_type_2

promo_id -> promo_type_1, promo_bin_1, promo_type_2, promo_bin_2, promo_discount_2, promo_discount_type_2

(Created separate tables for promotion for sets of values that apply to multiple rows.)

Product Details

{product_id, product_length, product_width, product_depth, cluster_id, hierarchy1_id,

hierarchy2_id, hierarchy3_id, hierarchy4_id, hierarchy5_id}

product_id → product_length, product_width, product_depth, cluster_id, hierarchy1_id,

hierarchy2_id, hierarchy3_id, hierarchy4_id, hierarchy5_id → **Follows BCNF**

hierarchy5_id gives the following ids in the promotion table.

{hierarchy5_id} → {hierarchy4_id, hierarchy3_id, hierarchy2_id, hierarchy1_id} →

{product_id, product_length, product_width, product_depth, cluster_id, hierarchy5_id}, {hierarchy1_id, hierarchy2_id, hierarchy3_id, hierarchy4_id, hierarchy5_id}

Violates BCNF because it is a transitive dependency; the elements on the right depend on the element hierarchy5_id, which is not a key.

hierarchy4_id → hierarchy3_id

hierarchy3_id → hierarchy2_id

hierarchy2_id → hierarchy1_id

{product_id, product_length, product_width, product_depth, cluster_id, hierarchy5_id},

{hierarchy1_id, hierarchy2_id}, {hierarchy2_id, hierarchy3_id}, {hierarchy3_id,

hierarchy4_id}, {hierarchy4_id, hierarchy5_id}.

Store Details:

store_id → storetype_id, store_size, city_id → **Follows BCNF**

d. The whole schema is not in BCNF because the product details section has transitive dependency. So, we will decompose:

1. Sales table into Promotion and Sales.
2. ProductDetails table into productDetails and Hierarchy table.

Now, we checked if the schema is in BCNF. A schema is in BCNF if for every non-trivial functional dependency, the left-hand side is a super key. A super key is a set of one or more attributes that can uniquely identify each tuple in a relation.

• In our case, we have the following tables in our schema:

1. Sales table
2. Product Details table
3. Store table
4. Hierarchy Table
5. Promotion Table

• Each of these tables satisfies the BCNF conditions, meaning that there are no non-trivial functional dependencies that violate the BCNF rules. As a result, there is no need to decompose these tables further.

The schema is now normalized to an acceptable degree, ensuring minimal redundancy, and reducing the risk of update anomalies.

Decomposition with the minimal keys:

Sales Table

● sales_id → product_id, store_id, date, sales, revenue, stock, price, promo_id

Product Details Table

● product_id → product_length, product_width, product_depth, cluster_id, hierarchy5_id

(Product table has a unique value of various products along with its dimensions.)

Store Details Table

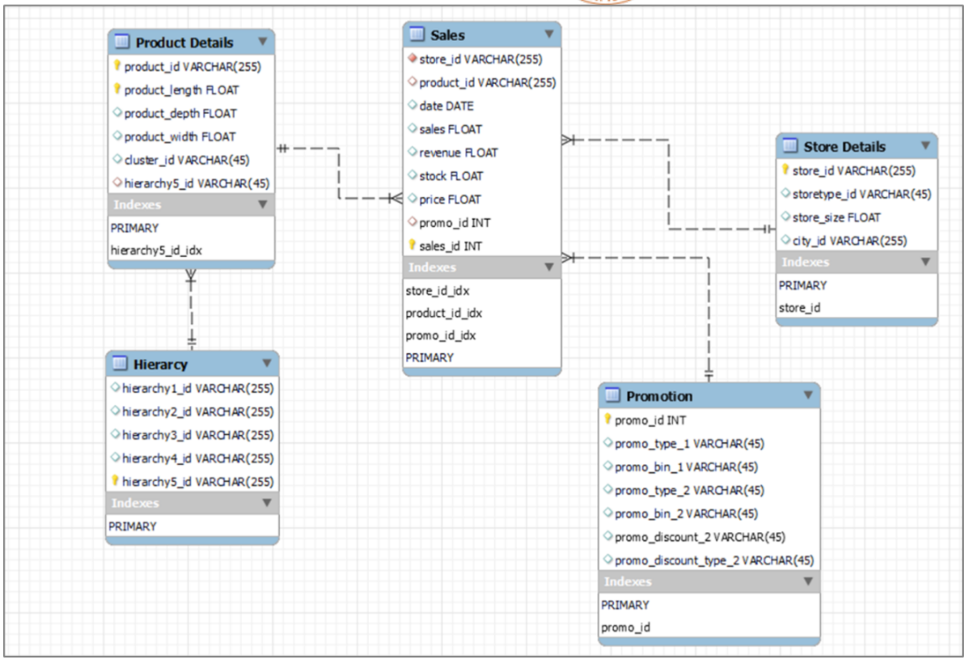
● store_id → storetype_id, store_size, city_id (This table provides the store type, its size and respective city.)

Hierarchy Table

● hierarchy5_id → hierarchy4_id, hierarchy3_id, hierarchy2_id, hierarchy1_id

Promotion Table

●promo_id → promo_type_1, promo_bin_1, promo_type_2, promo_bin_2, promo_discount_2, promo_discount_type_2 (This table now contains all the details of the promotion table along with its discount details.)



5. CREATING THE DATA BASE WITH THE NEW SCHEMA

- Creating a database from the new ER Diagram had some minor issues,
  - While creating product table following issue came:
    "Table storage engine for <TABLE> doesn't have this option on order by query (ERROR 1031)"
    Resolution: We resolved this by changing the ROW_FORMAT = FIXED to DEFAULT in the auto created code by the wizard.

  - Later while creating store details table we faced the following error:
    "Error 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ') INVISIBLE)
    ENGINE = InnoDB
    KEY_BLOCK_SIZE = 4' at line 10"
    Resolution: This was resolved by correcting the index statement by removing INVISIBLE from the statement.

  - Later while creating Promotion table we faced the following error:
    "Error 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''promo_id')
)ENGINE = InnoDB' at line 13"
    Resolution: We resolved this by correcting the index statement by removing VISIBLE from the statement.

6. IMPORTING THE DATA INTO THE TABLES

Before importing the data into MySQL workbench we did come data manipulation on R such as splitting the data into tables and added the unique identifiers in some tables.

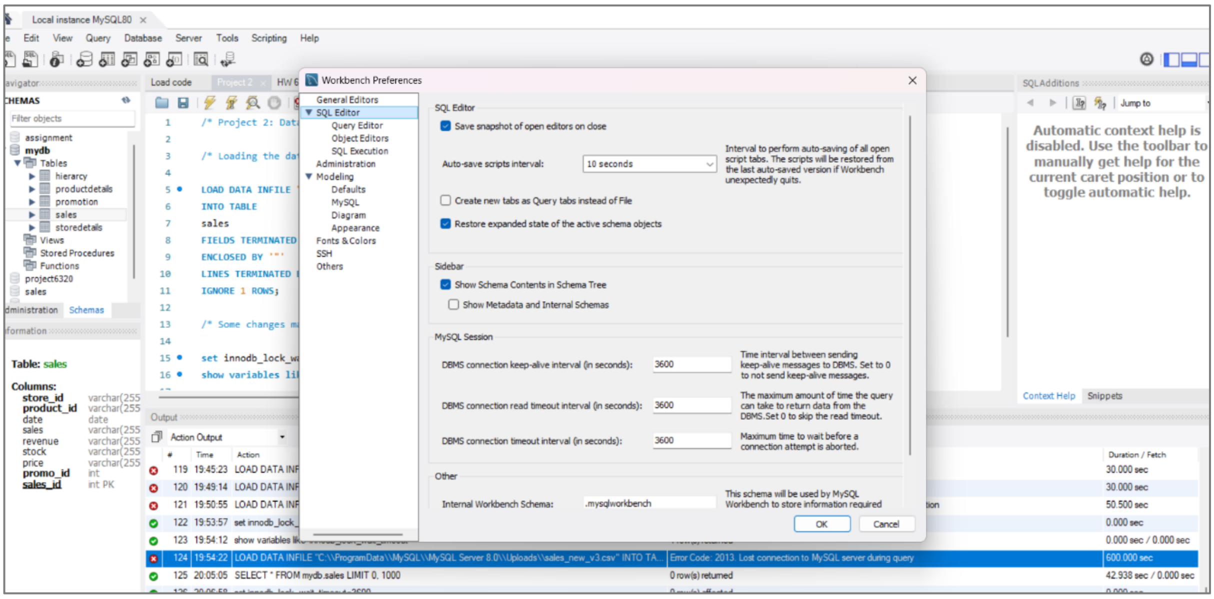
OTHER ERRORS THAT WE FACED:

1. We faced administrative and infile restriction while importing the error.
   Resolution: This issue was resolved by changing the location of data to uploads folder in MySQL folder of the system where it was given permission to load the files. Also, we had to use the “\\” in the path section of our code to resolve this issue.

2. “Error Code: 1262. Row 1 was truncated; it contained more data than there were input columns”
   Resolution: This was resolved by inserting the data via import wizard

3. “Error Code: 1265. Data truncated for column 'price' at row 1”
   While making changes with data in R, for importing there were NA values and they created problem while importing the data in MySQL workbench.
   Resolution: We resolved this issue by changing the numerical NA argument to 0 while exporting the data file from R and then loading the data in MySQL.

4. Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`mydb`.`sales`, CONSTRAINT `product_id` FOREIGN KEY (`product_id`) REFERENCES `product details` (`product_id`))
   While importing the sales data we had this issue which describes that the product_id does not exist in the productDetail table and hence it does not meet the foreign key constraints.
   Resolution: We resolved this error by checking the productDetails table and saw that SQL was removing some records with few NA entries and hence some records of product_id were missing because of the import by import wizard. We changed the NA in our data set to blank and then again imported the data in the table which eventually resolved the error.
5. Faced following timeout errors while uploading the sales data with load command.
   Error Code: 1205. Lock wait timeout exceeded; try restarting transaction
   Error Code: 2013. Lost connection to MySQL server during query
   Resolution: We resolved this by increasing the timeout interval and also by changing the “innodb_lock_wait_timeout” limit.



```
/* Project 2: Dat

/* Loading the da

LOAD DATA INFILE
INTO TABLE
sales
FIELDS TERMINATED
ENCLOSED BY '"'
LINES TERMINATED
IGNORE 1 ROWS;

/* Some changes m

set innodb_lock_w
show variables li
```

Table: sales

Columns:

store_id	varchar(255
product_id	varchar(255
date	date
sales	varchar(255
revenue	varchar(255
stock	varchar(255
price	varchar(255
promo_id	int
sales_id	int PK

SQL Editor

☑ Save snapshot of open editors on close

Auto-save scripts interval: 10 seconds — Interval to perform auto-saving of all open script tabs. The scripts will be restored from the last auto-saved version if Workbench unexpectedly quits.

☐ Create new tabs as Query tabs instead of File

☑ Restore expanded state of the active schema objects

Sidebar

☑ Show Schema Contents in Schema Tree

☐ Show Metadata and Internal Schemas

MySQL Session

DBMS connection keep-alive interval (in seconds): 3600 — Time interval between sending keep-alive messages to DBMS. Set to 0 to not send keep-alive messages.

DBMS connection read timeout interval (in seconds): 3600 — The maximum amount of time the query can take to return data from the DBMS.Set 0 to skip the read timeout.

DBMS connection timeout interval (in seconds): 3600 — Maximum time to wait before a connection attempt is aborted.

Other

Internal Workbench Schema: .mysqlworkbench — This schema will be used by MySQL Workbench to store information required

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
#    Time      Action                                                                                                                           Duration / Fetch
119  19:45:23  LOAD DATA INF                                                                                                                    30.000 sec
120  19:49:14  LOAD DATA INF                                                                                                                    30.000 sec
121  19:50:55  LOAD DATA INF                                                              tion                                                  50.500 sec
122  19:53:57  set innodb_lock_                                                                                                                 0.000 sec
123  19:54:12  show variables lik                                                                                                               0.000 sec / 0.000 sec
124  19:54:22  LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\sales_new_v3.csv" INTO TA...  Error Code: 2013. Lost connection to MySQL server during query  600.000 sec
125  20:05:05  SELECT * FROM mydb.sales LIMIT 0, 1000                                    0 row(s) returned                                     42.938 sec / 0.000 sec
```

SECTION 2: DATA CLEANING AND DATABASE TESTING

1. ***For each table in your database, check all the columns and the values they contain***

   After updating the new schema we have the following tables and attributes (along with their data types)

Table Name	Attribute	Attribute type
StoreDetails	store_id	VARCHAR(255)
	storetype_id	VARCHAR(45)
	store_size	FLOAT
	city_id	VARCHAR(255)
Promotion	promo_id	VARCHAR(45)
	promo_type_1	VARCHAR(45)
	promo_bin_1	VARCHAR(45)
	promo_type_2	VARCHAR(45)
	promo_bin_2	VARCHAR(45)
	promo_discount_2	VARCHAR(45)
	promo_discount_type_2	VARCHAR(45)
ProductDetails	product_id	VARCHAR(255)
	product_length	FLOAT
	product_depth	FLOAT
	product_width	FLOAT

Table Name	Attribute	Attribute type
	cluster_id	VARCHAR(45)
	hierarchy5_id	VARCHAR(45)
Hierarchy	hierarchy1_id	VARCHAR(255)
	hierarchy2_id	VARCHAR(255)
	hierarchy3_id	VARCHAR(255)
	hierarchy4_id	VARCHAR(255)
	hierarchy5_id	VARCHAR(255)
Sales	store_id	VARCHAR(255)
	product_id	VARCHAR(255)
	date	DATE
	sales	FLOAT
	revenue	FLOAT
	stock	FLOAT
	price	FLOAT
	promo_id	INT
	sales_id	INT

2. ***Check them against the information you found in step 3 of project 1***

While uploading the sales data in our table in MySQL workbench we had to do some modification with NA values and replace it with 0 for importing.The Statistics for the numerical attributes are approximately the same what mentioned in step 3 of project 1. The values of these statistical measures are mentioned in the query log attached below.

Table	Variable	Mean	Standard Deviation
Sales	Sales	0.41	14.21
Sales	Revenue	1.94	38.66
Sales	Stock	14.91	36.32
Sales	Price	15.13	31.93

3. ***Checking for data discrepancies:***

a. **Discrepancy:** In productDetails table we have some records showing product length, depth, and width as 0, which we are assuming are blanks because while importing the data the NAs were converted as zeros.
Resolution: We will replace these zeros with the average values of the attributes, and we will be taking the average values with respect to non zeros

b. **Discrepancy:** In promotion table while importing the data we imported the NAs in product_discount_2 column
Resolution: We changed these values to NULL by using update command

4. ***Make sure all the values of these columns are from the same type (all numeric)***

   After reviewing all the character variables in all the data tables, everything is correct, and all the columns are from the same data type as per our schema.

QUERY LOGS FOR IMPORT AND DATA CHECK / APPENDIX:

```
/* Project 2: Data Base Foundation for Business Analytics */

/* Loading the data in sales table*/

LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server
8.0\\Uploads\\sales_new_v5.csv"
INTO TABLE
sales
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

/* Some changes made temporarily to change the timeout time to upload the data */

set innodb_lock_wait_timeout=3600;
show variables like 'innodb_lock_wait_timeout';

/*
=================================================================
===================================*/

/* Checking for data discrepancies*/

/* Checking for data discrepancies in productdetails table*/

SELECT *
FROM productdetails;

SELECT *
FROM productdetails
WHERE product_length = 0;

UPDATE productdetails
SET product_length = (
SELECT AVG(product_length)
FROM productdetails WHERE product_length != 0)
WHERE product_length = 0;


SELECT AVG(product_length)
```

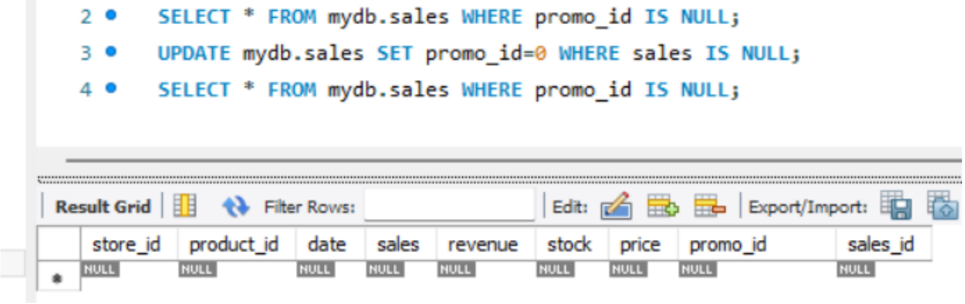
```
FROM productdetails
WHERE product_length <> 0;

UPDATE productdetails
SET product_length = 7.243
WHERE product_length = 0;

SELECT AVG(product_width)
FROM productdetails
WHERE product_width <> 0;

UPDATE productdetails
SET product_width = 13.474
WHERE product_width = 0;

SELECT AVG(product_depth)
FROM productdetails
WHERE product_depth <> 0;

UPDATE productdetails
SET product_depth = 18.478
WHERE product_depth = 0;

/* Checking for data discrepancies in hierarchy table: No numerical variables in this table*/
/* Checking for leading and trailing spces in hierarchy table*/

SELECT *
FROM hierarcy
WHERE hierarchy1_id LIKE ' %' or hierarchy1_id LIKE '% ';

SELECT *
FROM hierarcy
WHERE hierarchy2_id LIKE ' %' or hierarchy2_id LIKE '% ';

SELECT *
FROM hierarcy
WHERE hierarchy3_id LIKE ' %' or hierarchy3_id LIKE '% ';

SELECT *
FROM hierarcy
WHERE hierarchy4_id LIKE ' %' or hierarchy4_id LIKE '% ';

SELECT *
FROM hierarcy
WHERE hierarchy5_id LIKE ' %' or hierarchy5_id LIKE '% ';


/* Checking for data discrepancies in promotion table*/

SELECT *
```

```
FROM promotion;

UPDATE promotion
SET promo_discount_2 = NULL
WHERE promo_discount_2 = "NA";


/* Checking for data discrepancies in storedetails table*/

SELECT *
FROM storedetails;

SELECT *
FROM storedetails
WHERE store_size IS NULL;

/* storedetail table seems fine as there are no missing values in the numeric data column i.e.
store_size */


/* Chcking the discrepancies in sales table */

SELECT AVG(sales)
FROM sales;

/* mean = 0.41 */

SELECT stddev(sales)
FROM sales;

/* Standard Dev = 14.21 */

SELECT AVG(revenue)
FROM sales;
/* mean = 1.94 */
SELECT stddev(revenue)
FROM sales;
/* Standard Dev = 38.66 */
SELECT AVG(stock)
FROM sales;
/* mean = 14.91 */
SELECT stddev(stock)
FROM sales;
/* Standard Dev = 36.32 */
SELECT AVG(price)
FROM sales;
/* mean = 15.13 */
SELECT stddev(price)
FROM sales;
/* Standard Dev = 31.93 */
```

DATA CHECK FOR CHARACTER VARIABLES

1. *Check them against the information you found in step 3 of project 1.*

For the character columns, a few of the columns in the sales table contained large number of null values and removed those majorly from the promotion column. As shown in section 2-part 1 table, All the promotion columns are of character table.



```
SELECT * FROM mydb.sales WHERE promo_id IS NULL;
UPDATE mydb.sales SET promo_id=0 WHERE sales IS NULL;
SELECT * FROM mydb.sales WHERE promo_id IS NULL;
```

store_id	product_id	date	sales	revenue	stock	price	promo_id	sales_id
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. *Checking for data discrepancies:*

For removing discrepancies, separated out the unused products from the product_details table and deleted incomplete or missing data from the sales table. The following query was performed in order to remove discrepancies.

a. **Discrepancy:** In promo_bin_1 column contains missing values in the sales table which are of character type.

   Resolution: For removing the white space and null values in the promo_bin_1 column we are separating the whole column and making a new table for it. So, by doing these, we can remove the anomalies in the sales table

b. **Discrepancy:** cluster_id column contains missing values in the product_hierarchy table which are of character type.

   Resolution: For removing the white space in the cluster_id column, we are querying in the particular cluster_id column to remove all the null values. So, by doing these, we can remove the anomalies in the product_hierarchy table.

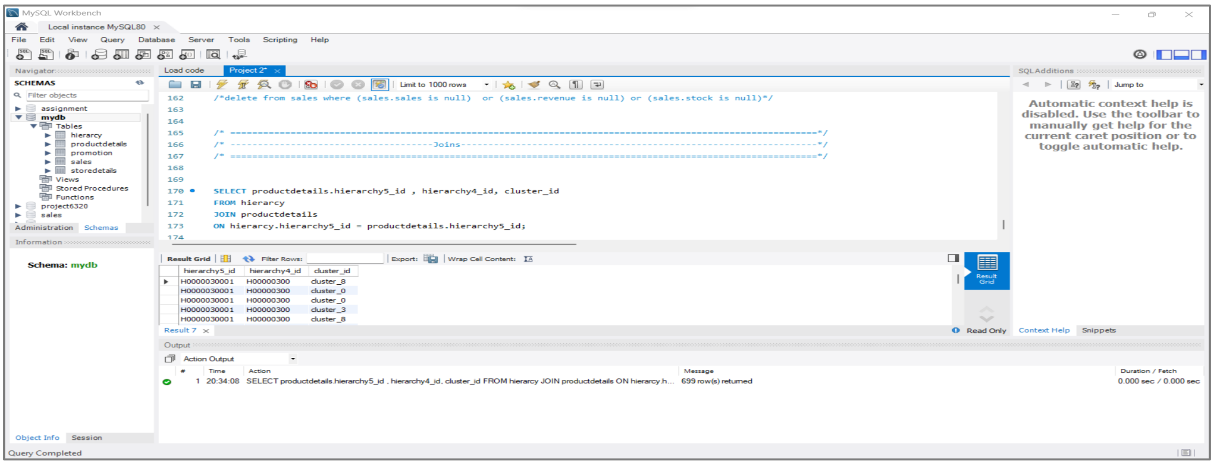
3. ***Make sure all the values of these columns are from the same type (all character)***
   After reviewing all the numerical variables in all the data tables, everything is correct, and all the columns are from the same data type as per our schema.

QUERY LOGS FOR CHARATER VARIABLES DATA CHECK / APPENDIX:

```
CREATE TABLE unused_products as SELECT productdetails.*
FROM productdetails
LEFT JOIN sales
ON productdetails.product_id = sales.product_id
where sales.product_id IS NULL;

/* to delete the null values */
DELETE FROM sales where (sales.sales IS NULL) OR (sales.revenue is NULL) OR
(sales.stock IS
NULL)
```

APPLYING JOINS TO CHECK WHETHER JOINS WORK PROPERLY IN OUR NEW SCHEMA:

Join 1:



```
/*delete from sales where (sales.sales is null)  or (sales.revenue is null) or (sales.stock is null)*/


/* ==========================================================================================*/
/* -----------------------------------Joins---------------------------------------------------*/
/* ==========================================================================================*/


SELECT productdetails.hierarchy5_id , hierarchy4_id, cluster_id
FROM hierarcy
JOIN productdetails
ON hierarcy.hierarchy5_id = productdetails.hierarchy5_id;
```

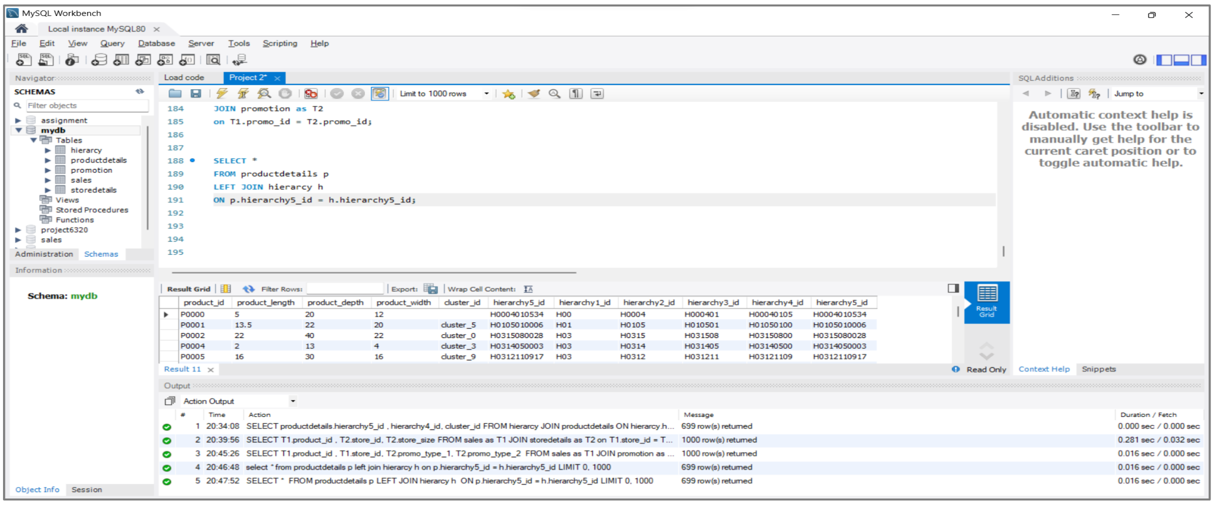
hierarchy5_id	hierarchy4_id	cluster_id
H000030001	H00000300	cluster_8
H000030001	H00000300	cluster_0
H000030001	H00000300	cluster_0
H000030001	H00000300	cluster_3
H000030001	H00000300	cluster_8

Schema: mydb

```
#  Time      Action                                                                                              Message               Duration / Fetch
1  20:34:08  SELECT productdetails.hierarchy5_id , hierarchy4_id, cluster_id FROM hierarcy JOIN productdetails ON hierarcy.h...  699 row(s) returned  0.000 sec / 0.000 sec
```

Query Completed

Join 2:



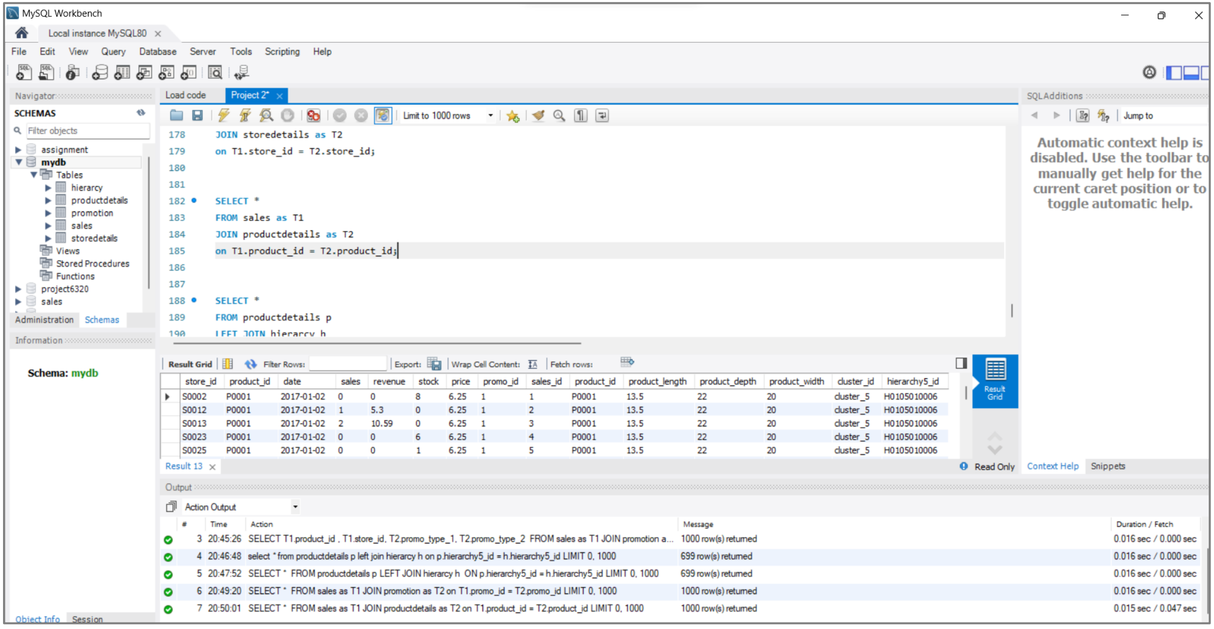
```
JOIN promotion as T2
on T1.promo_id = T2.promo_id;


SELECT *
FROM productdetails p
LEFT JOIN hierarcy h
ON p.hierarchy5_id = h.hierarchy5_id;
```

product_id	product_length	product_depth	product_width	cluster_id	hierarchy5_id	hierarchy1_id	hierarchy2_id	hierarchy3_id	hierarchy4_id	hierarchy5_id
P0000	5	20	12		H0004010534	H00	H0004	H000401	H00040105	H0004010534
P0001	13.5	22	20	cluster_5	H0105010006	H01	H0105	H010501	H01050100	H0105010006
P0002	22	40	22	cluster_0	H0315080028	H03	H0315	H031508	H03150800	H0315080028
P0004	2	13	4	cluster_3	H0314050003	H03	H0314	H031405	H03140500	H0314050003
P0005	16	30	16	cluster_9	H0312110917	H03	H0312	H031211	H03121109	H0312110917

Schema: mydb

```
#  Time      Action                                                                                                  Message                Duration / Fetch
1  20:34:08  SELECT productdetails.hierarchy5_id , hierarchy4_id, cluster_id FROM hierarcy JOIN productdetails ON hierarcy.h...  699 row(s) returned   0.000 sec / 0.000 sec
2  20:39:56  SELECT T1.product_id , T2.store_id, T2.store_size FROM sales as T1 JOIN storedetails as T2 on T1.store_id = T...  1000 row(s) returned  0.281 sec / 0.032 sec
3  20:45:26  SELECT T1.product_id , T1.store_id, T2.promo_type_1, T2.promo_type_2  FROM sales as T1 JOIN promotion as ...    1000 row(s) returned  0.016 sec / 0.000 sec
4  20:46:48  select * from productdetails p left join hierarcy h on p.hierarchy5_id = h.hierarchy5_id LIMIT 0, 1000          699 row(s) returned   0.016 sec / 0.000 sec
5  20:47:52  SELECT *  FROM productdetails p LEFT JOIN hierarcy h  ON p.hierarchy5_id = h.hierarchy5_id LIMIT 0, 1000        699 row(s) returned   0.016 sec / 0.000 sec
```

Join 3:



```
JOIN storedetails as T2
on T1.store_id = T2.store_id;


SELECT *
FROM sales as T1
JOIN productdetails as T2
on T1.product_id = T2.product_id;


SELECT *
FROM productdetails p
LEFT JOIN hierarcy h
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

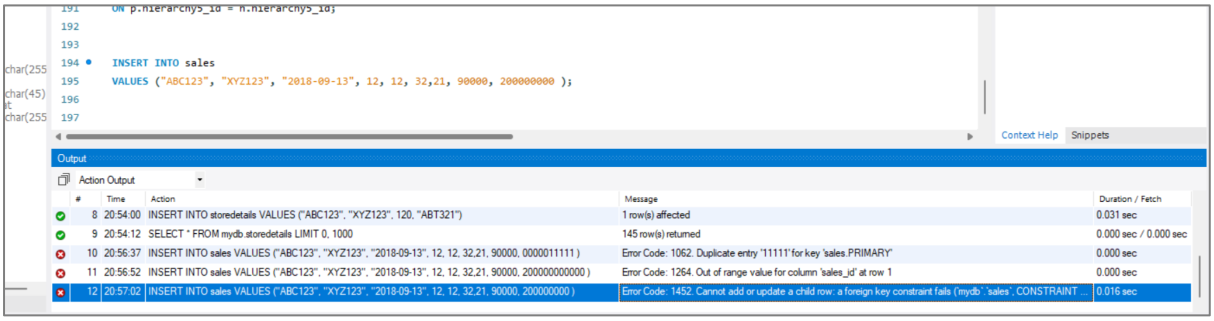
store_id	product_id	date	sales	revenue	stock	price	promo_id	sales_id	product_id	product_length	product_depth	product_width	cluster_id	hierarchy5_id
S0002	P0001	2017-01-02	0	0	8	6.25	1	1	P0001	13.5	22	20	cluster_5	H0105010006
S0012	P0001	2017-01-02	1	5.3	0	6.25	1	2	P0001	13.5	22	20	cluster_5	H0105010006
S0013	P0001	2017-01-02	2	10.59	0	6.25	1	3	P0001	13.5	22	20	cluster_5	H0105010006
S0023	P0001	2017-01-02	0	0	6	6.25	1	4	P0001	13.5	22	20	cluster_5	H0105010006
S0025	P0001	2017-01-02	0	0	1	6.25	1	5	P0001	13.5	22	20	cluster_5	H0105010006

```
#  Time      Action                                                                                          Message                Duration / Fetch
3  20:45:26  SELECT T1.product_id , T1.store_id, T2.promo_type_1, T2.promo_type_2  FROM sales as T1 JOIN promotion a...  1000 row(s) returned  0.016 sec / 0.000 sec
4  20:46:48  select * from productdetails p left join hierarcy h on p.hierarchy5_id = h.hierarchy5_id LIMIT 0, 1000       699 row(s) returned   0.016 sec / 0.000 sec
5  20:47:52  SELECT *  FROM productdetails p LEFT JOIN hierarcy h  ON p.hierarchy5_id = h.hierarchy5_id LIMIT 0, 1000     699 row(s) returned   0.016 sec / 0.000 sec
6  20:49:20  SELECT *  FROM sales as T1 JOIN promotion as T2 on T1.promo_id = T2.promo_id LIMIT 0, 1000                   1000 row(s) returned  0.016 sec / 0.000 sec
7  20:50:01  SELECT *  FROM sales as T1 JOIN productdetails as T2 on T1.product_id = T2.product_id LIMIT 0, 1000         1000 row(s) returned  0.015 sec / 0.047 sec
```

QUERIES TO CHECK FOR CONSTRAINTS:

Constraints are also working as shown in the below queries.

Query 1

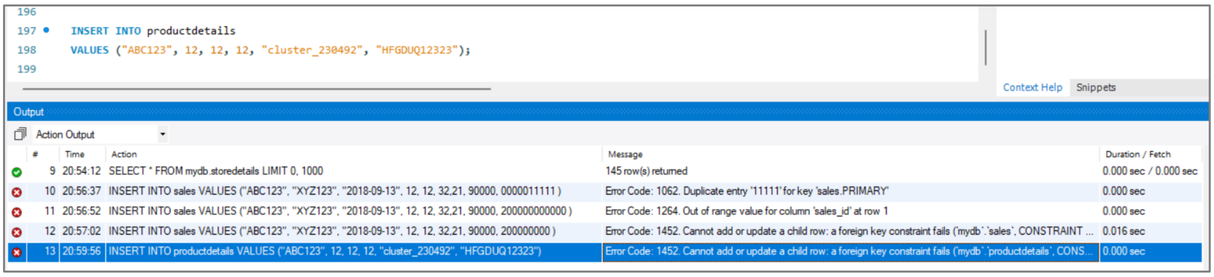


```
ON p.hierarchy5_id = h.hierarchy5_id;


INSERT INTO sales
VALUES ("ABC123", "XYZ123", "2018-09-13", 12, 12, 32,21, 90000, 200000000 );
```

```
#   Time      Action                                                                                            Message                                                                                         Duration / Fetch
8   20:54:00  INSERT INTO storedetails VALUES ("ABC123", "XYZ123", 120, "ABT321")                               1 row(s) affected                                                                                0.031 sec
9   20:54:12  SELECT * FROM mydb.storedetails LIMIT 0, 1000                                                     145 row(s) returned                                                                              0.000 sec / 0.000 sec
10  20:56:37  INSERT INTO sales VALUES ("ABC123", "XYZ123", "2018-09-13", 12, 12, 32,21, 90000, 0000011111 )    Error Code: 1062. Duplicate entry '11111' for key 'sales.PRIMARY'                                0.000 sec
11  20:56:52  INSERT INTO sales VALUES ("ABC123", "XYZ123", "2018-09-13", 12, 12, 32,21, 90000, 200000000000 )  Error Code: 1264. Out of range value for column 'sales_id' at row 1                             0.000 sec
12  20:57:02  INSERT INTO sales VALUES ("ABC123", "XYZ123", "2018-09-13", 12, 12, 32,21, 90000, 200000000 )     Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`mydb`.`sales`, CONSTRAINT ...  0.016 sec
```

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`mydb`.`sales`, CONSTRAINT `product_id` FOREIGN KEY (`product_id`) REFERENCES `productdetails` (`product_id`))

Query 2

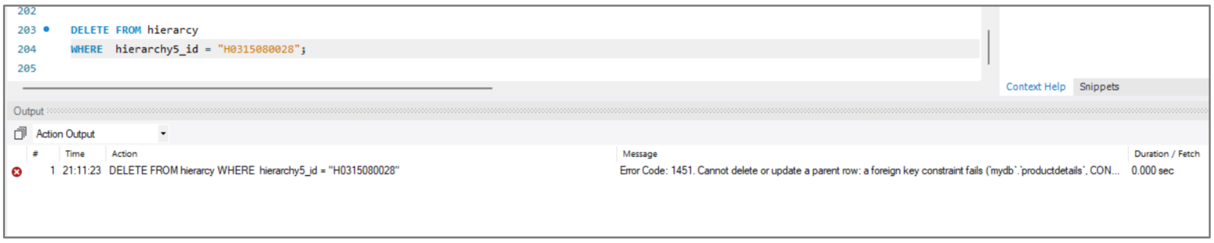


```
INSERT INTO productdetails
VALUES ("ABC123", 12, 12, 12, "cluster_230492", "HFGDUQ12323");
```

```
#   Time      Action                                                                                            Message                                                                                                 Duration / Fetch
9   20:54:12  SELECT * FROM mydb.storedetails LIMIT 0, 1000                                                     145 row(s) returned                                                                                      0.000 sec / 0.000 sec
10  20:56:37  INSERT INTO sales VALUES ("ABC123", "XYZ123", "2018-09-13", 12, 12, 32,21, 90000, 0000011111 )    Error Code: 1062. Duplicate entry '11111' for key 'sales.PRIMARY'                                        0.000 sec
11  20:56:52  INSERT INTO sales VALUES ("ABC123", "XYZ123", "2018-09-13", 12, 12, 32,21, 90000, 200000000000 )  Error Code: 1264. Out of range value for column 'sales_id' at row 1                                     0.000 sec
12  20:57:02  INSERT INTO sales VALUES ("ABC123", "XYZ123", "2018-09-13", 12, 12, 32,21, 90000, 200000000 )     Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`mydb`.`sales`, CONSTRAINT ...      0.016 sec
13  20:59:56  INSERT INTO productdetails VALUES ("ABC123", 12, 12, 12, "cluster_230492", "HFGDUQ12323")         Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`mydb`.`productdetails`, CONS...  0.000 sec
```

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`mydb`.`productdetails`, CONSTRAINT `hierarchy5_id` FOREIGN KEY (`hierarchy5_id`) REFERENCES `hierarcy` (`hierarchy5_id`))

Query 3



```sql
DELETE FROM hierarcy
WHERE  hierarchy5_id = "H0315080028";
```

```
#  Time      Action                                                  Message                                                                                          Duration / Fetch
1  21:11:23  DELETE FROM hierarcy WHERE  hierarchy5_id = "H0315080028"  Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('mydb'.'productdetails', CON...  0.000 sec
```

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`mydb`.`productdetails`, CONSTRAINT `hierarchy5_id` FOREIGN KEY (`hierarchy5_id`) REFERENCES `hierarcy` (`hierarchy5_id`))

***** End of Document*****