Practical Assignment 1
Analysis of Data Sources

Group Number: 36

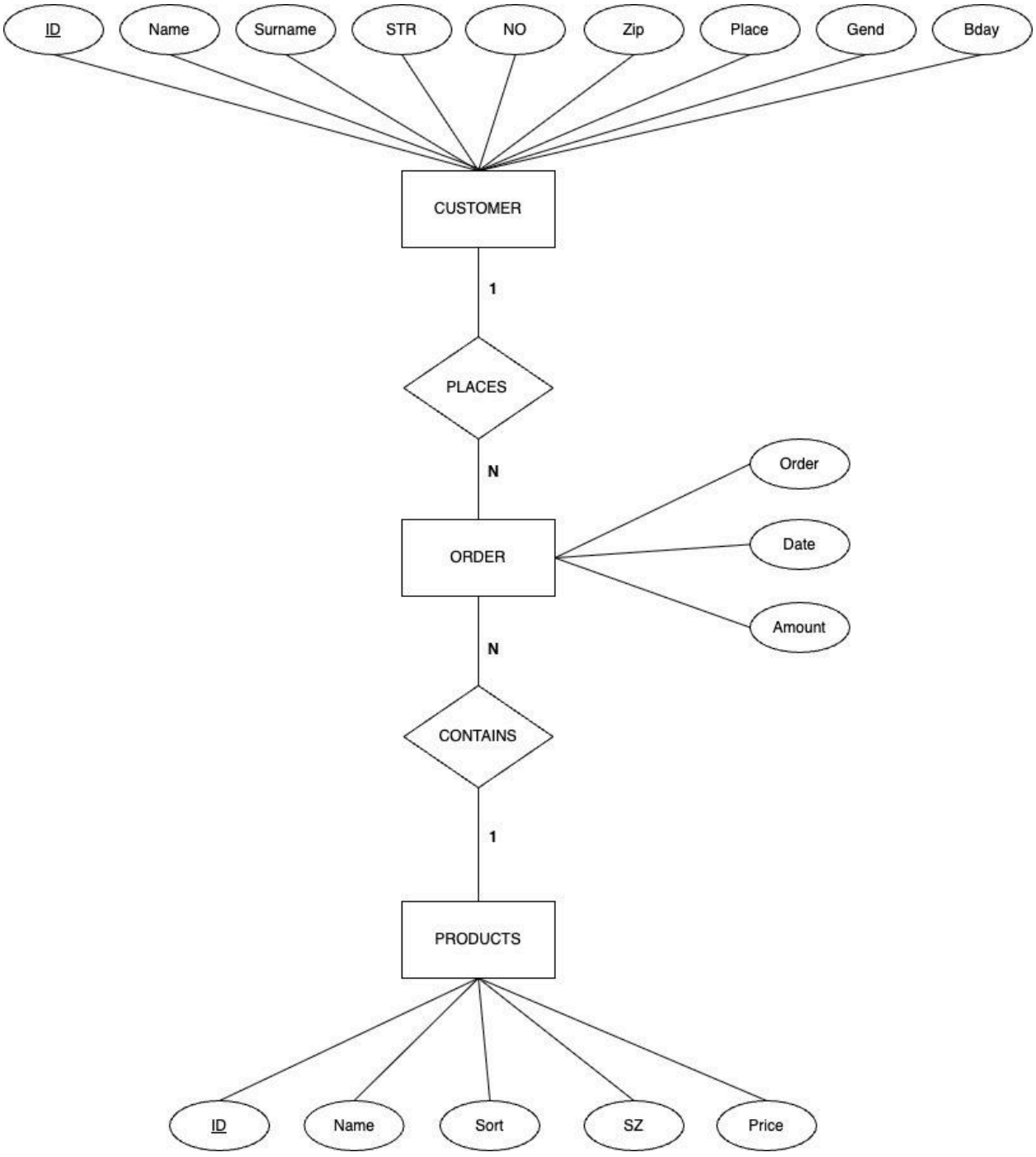
Branches assigned to us:

BRANCH_1
BRANCH_2
BRANCH_5

2. The queries we used to determine the table structures are present in the **analysis.sql** file

3. The DDL statements/ CREATE statement are present in the **create.sql** file

OBSERVED E/R SCHEMA OF BRANCH_1:



Observed ambiguity, design issues in branch 1:

Orders table references both customer table and product table, we have named the relationship between them as “places” and “contains”. References are not named in the DB.

Orders table:

- There is an attribute **ORDER**. This value is too ambiguous, something like **ORDER_ID** would have been more fitting and it should have been a primary key.

- Also, **ORDER** is a keyword in sql and it is not ideal to name a column as **ORDER**. The **DATE, CSTM, PRO** have a combined unique constraint. A customer can buy the same product more than once on the same date.

- **AMOUNT** attribute has a **CHAR** data type, it should ideally be a number.

Customer table:

- The attributes for the customer’s location can be better grouped, a separate table for the location would have been better.

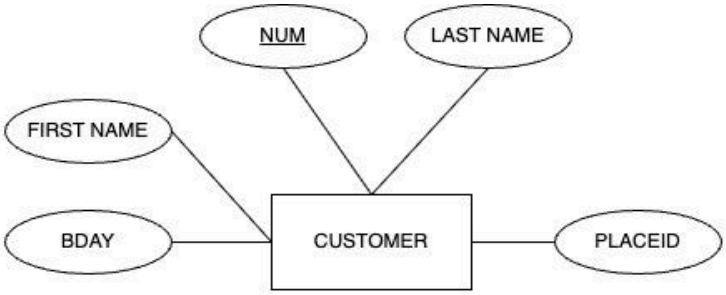
- **GEND** attribute should have a better defined constraint to allow only specific values.

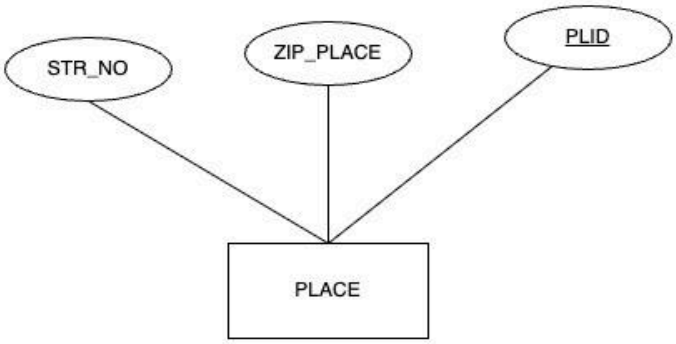
Products table:

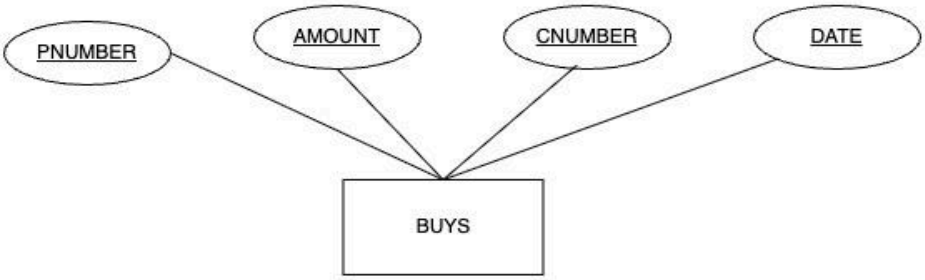
- The attribute **SZ** is too vague, the naming could have been better.

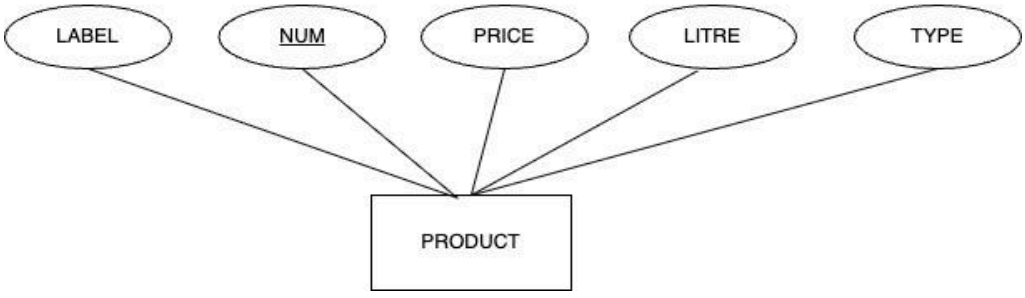
- The data length on **PRICE** is too limiting.

OBSERVED E/R SCHEMA OF BRANCH_2:









Observed ambiguity, design issues in branch 2:

Overall, there is no well defined relationship, foreign keys to combine the tables.

1. Product table:

- COLUMN name of NUM is not clear and used as Primary key in this Product Table
- Renaming NUM as PRODUCT_ID it is easier to understand the Table.
- The TYPE column allows any text, users can enter any values this may take into errors in queries.
- Either add a CHECK constraint or create a separate PRODUCT_TYPE table for reference.
- To improve better execution in LABEL or TYPE columns with using the WHERE clauses or JOIN.
- Change the data type of the PRICE column from TEXT (VARCHAR2) to NUMBER(10,3) to ensure correct output.

2. Customer Table

- COLUMN name of NUM is not clear and used as Primary key in this Customer Table
- Renaming NUM as CUSTOMER_ID, it is easier to understand the Table.
- The table says both the FIRSTNAME and LASTNAME must be unique. But FIRSTNAME can be empty(NULL) can allow duplicate entries which disturbs the Table.
- Either make FIRSTNAME can not be empty or remove it from the table.
- Allow dates to be realistic - like after 1900 will be better.

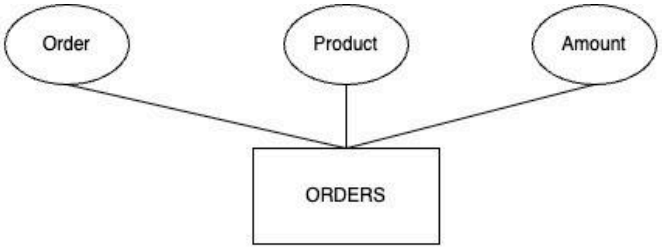
3. Place Table

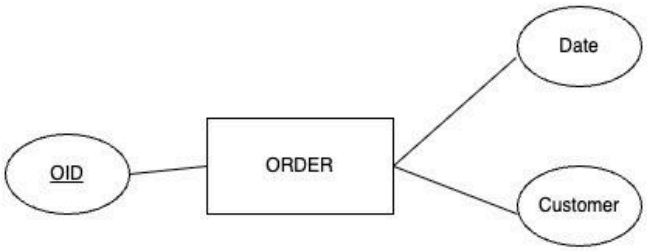
- COLUMN name of PLID is not clear and used as Primary key in this Place Table
- Renaming PLID as PLACE_ID it is easier to understand the Table.
- Column ZIP_PLACE stores both zip code and city together which is not ideal. Separate this into two columns states that one for ZIP and other for CITY
- Column STR_NO is it related to street or building number can define what STR_NO or split into STREET_NAME,BULDING_NUMBER

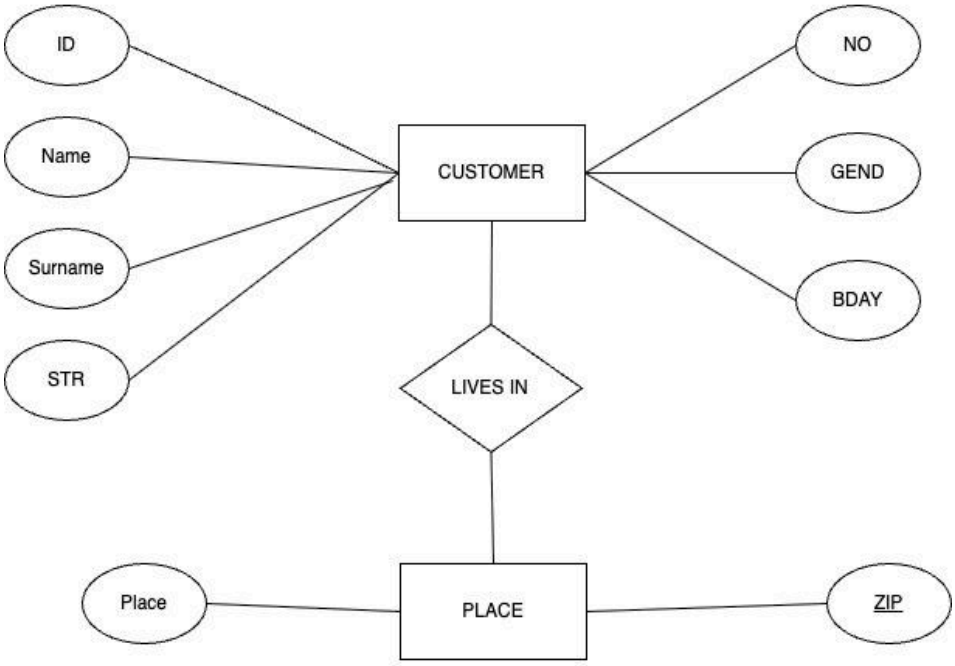
4. Buys table:

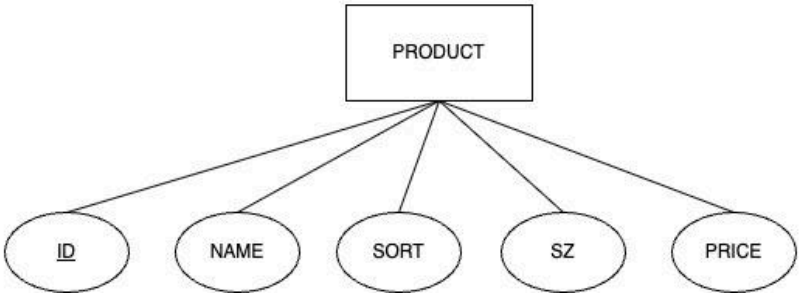
- In this table Primary key is defined as a combination of CNUMBER, PNUMBER, AMOUNT and DATE. But AMOUNT is not something that uniquely defines and it breaks the principle of good database design.
- It’s better to introduce a unique-PAYMENT_ID
- There is no Foreign key that represents the customer and product table.
- Change the data type of the AMOUNT column from TEXT (VARCHAR2) to NUMBER(10,3) to ensure correct output.
- Column DATE used as a special keyword in SQL.This can confuse the Database.
- Renaming it to PURCHASE_DATE is something clearer.

OBSERVED E/R SCHEMA OF BRANCH_5:


Order
Product
Amount
ORDERS


Date
OID
ORDER
Customer


ID
NO
Name
CUSTOMER
GEND
Surname
BDAY
STR
LIVES IN
Place
PLACE
ZIP


PRODUCT
ID
NAME
SORT
SZ
PRICE

Observed ambiguity, design issues in branch 5:

- Overall, similar to branch 2, there is no well defined relationship in the schema. **Lack of foreign keys** references in all tables.

ORDERS/ORDER table:

- These two tables are not well defined.
- Usage of the keyword "**ORDER**" does not seem ideal. **ORDER_ID** would have been a better name.
- No primary key in the **ORDERS** table and does not reference the **ORDER** table also.
- "**DATE**" attribute is also problematic since it is the name of the key in sql.
- The **AMOUNT** attribute limits the value to 0-99, it is better to have something like (5,0) instead of (2,0).

PRODUCT table:

- Product table has a unique key but not a primary key.
- The attribute **SZ** is vague, hard to identify its purpose.
- The **NAME** attribute should have been ideally **NOT NULL**.

CUSTOMER table:

- No primary key defined for this table, **ID** value should have been a primary key.
- **GEN** attribute could have a well defined constraint to allow only certain values.