2023

Checkpoint Technical Activity

LOU GEH SUPERMARKET

BFI

Table of Contents

Diagram

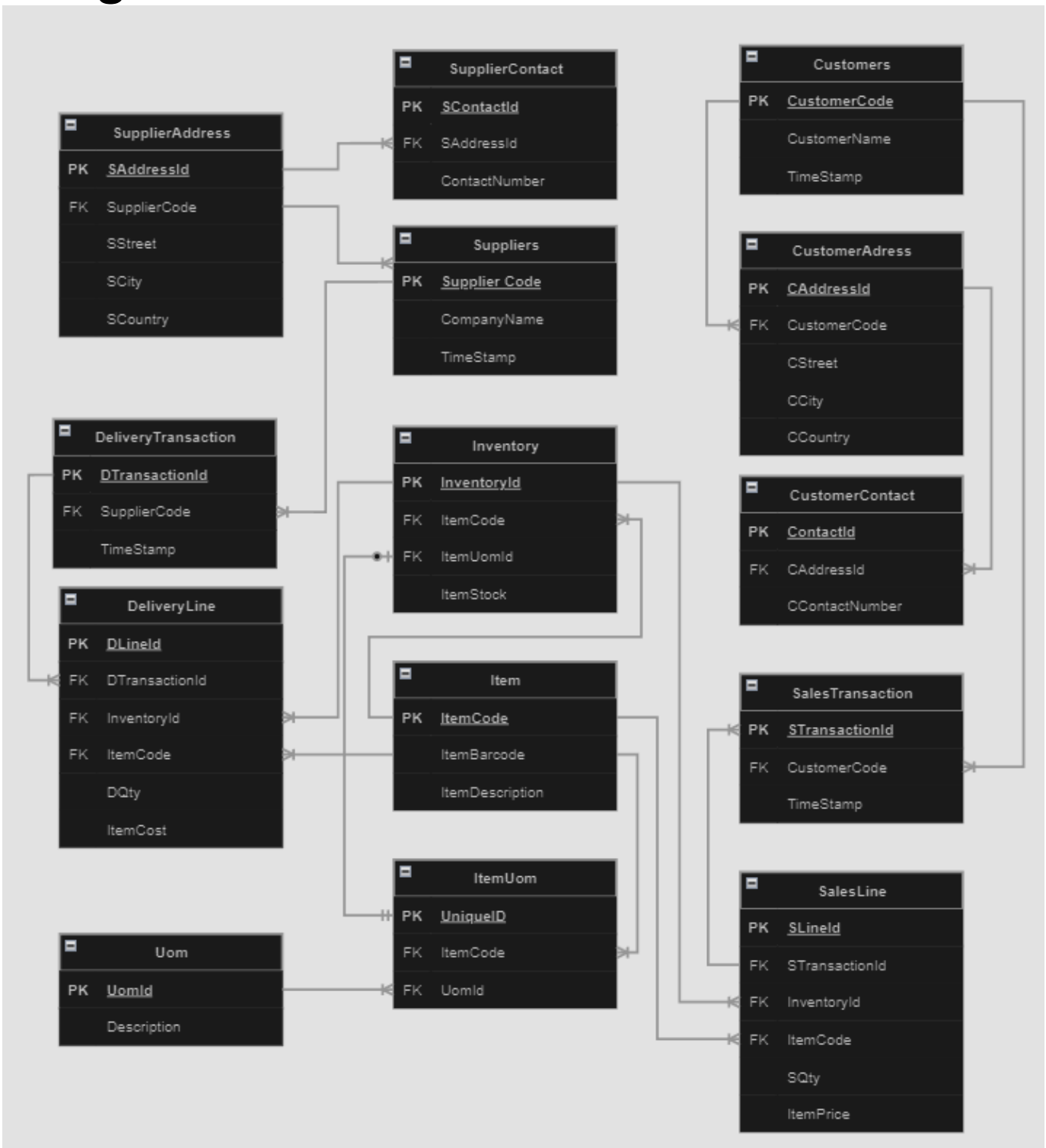

Figure 1 Entity Relationship Diagram of Lou Geh Supermarket

Normalization

Third Normal Form

The database is in Third Normal Form (3NF) since it satisfies the following:

- It is in Second Normal Form (2NF)

  This means that it has a primary key, and all non-key attributes (columns) are functionally dependent on the entire primary key, eliminating partial dependencies.

- It does not have any transitive dependencies

  Transitive dependencies occur when non-key attributes depend on other non-key attributes within the same table.

Indexing

By indexing columns, you can improve the performance of queries that use them. The index will allow the database to quickly find the rows that match your query criteria, without having to scan the entire table.

Suppliers	**DeliveryTransaction**	**SalesTransaction**	**Inventory**
-SupplierCode	-DTransactionId	-STransactionId	-InventoryId
-CompanyName	-SupplierCode	-CustomerCode	-ItemCode
Customers	**DeliveryLine**	**SalesLine**	**Item**
-CustomerCode	-DLineId	-SLineId	-ItemCode
-CustomerName	-DTransactionId	-StransactionId	-ItemBarcode
	-ItemCode	-ItemCode	

These columns are often used in joins and filters, so indexing them can also improve the performance of queries. For example, if you query the *Customers* table by *CustomerName*, the database can use the index on the *CustomerName* column to quickly find the row that matches your criteria. Without an index, the database would have to scan the entire *Customers* table to find the matching row.

```
CREATE INDEX idx_CompanyName ON Suppliers (CompanyName);
CREATE INDEX idx_CustomerName ON Customers (CustomerName);
```

Attributes and Keys

In SQL, an attribute is a column or field in a database table. It represents a specific characteristic or property of the data stored in that table. Attributes define the types of information that can be stored in a table.

Suppliers
-SupplierCode, *SERIAL PK*
-CompanyName *varchar*
-TimeStamp, *timestamp*

Customers
-CustomerCode, *SERIAL PK*
-CustomerName *varchar*
-TimeStamp *timestamp*

Inventory
-InventoryId, *SERIAL PK*
-ItemCode, *integer FK*
-ItemUomId, *integer FK*
-ItemStock *integer*

SupplierAddress
-SAddressId, *SERIAL PK*
-SupplierCode, *integer FK*
-SStreet *varchar*
-SCity *varchar*
-SCountry *varchar*

CustomerAddress
-CAddressId, *SERIAL PK*
-CustomerCode, *integer FK*
-CStreet *varchar*
-CCity *varchar*
-CCountry *varchar*

Item
-ItemCode, *SERIAL PK*
-ItemBarcode *varchar*
-ItemDescription *varchar*

SupplierContact
-SContactId, *SERIAL PK*
-SAddressId, *integer FK*
-ContactNumber *varchar*

CustomerContact
-ContactId, *SERIAL PK*
-CAddressId, *integer FK*
-CContactNumber *varchar*

ItemUom
-UniqueID, *SERIAL PK*
-ItemCode, *integer FK*
-UomId, *integer FK*

DeliveryTransaction
-DTransactionId, *SERIAL PK*
-SupplierCode, *integer FK*
-TimeStamp *timestamp*

SalesTransaction
-STransactionId, *SERIAL PK*
-CustomerCode, *integer FK*
-TimeStamp *timestamp*

Uom
-UomId, *SERIAL PK*
-Description *varchar*

DeliveryLine
-DLineId, *SERIAL PK*
-DTransactionId, *integer FK*
-InventoryId, *integer FK*
-ItemCode, *integer FK*
-DQty *integer*
-ItemCost *decimal(10,2)*

SalesLine
-SLineId, *SERIAL PK*
-StransactionId, *integer FK*
-InventoryId, *integer FK*
-ItemCode, *integer FK*
-SQty *integer*
-ItemPrice *decimal(10,2)*

```
ALTER TABLE Suppliers ADD CONSTRAINT SupplierCode PRIMARY KEY (SupplierCode);

ALTER TABLE SuppliersAddress ADD CONSTRAINT SupplierCode FOREIGN KEY (SupplierCode)
REFERENCES Suppliers;
```