

Table of Content

1.0 Introduction	3
1.1 Description of The Organization	3
1.2 Description of The Current System and Its Problem	3
1.3 Motivation	4
1.4 How will the organization gain from the integration of this system?	4
1.5 Module	4
2.0 User Requirement and Business Rules	7
2.1 Customer Module	7
2.2 Employee Module	7
2.3 Sales Module	8
2.4 Inventory Module	9
2.5 Cross Module User Requirements and Business Rules	10
3.0 Entity Relationship Modelling	10
3.1 Customer Module	11
3.2 Employee Module	13
3.3 Sales Module	16
3.4 Inventory Module	19
3.5 Combine ERD AIU SHOPS	22
4.0 Normalization	25
4.1 Customer Module	25
4.2 Employee Module	26
4.3 Sales Module	26
4.4 Inventory Module	28
5.0 Data dictionary	29
5.1 Customer Module	29
5.2 Employee Module	29
5.3 Sales Module	30
5.4 Inventory Module	30
6.0 Database Implementation	31
7.0 Front-end System Design and Implementation.	38
8.0 Project Problems and Pitfalls	43
9.0 Conclusion/Recommendations/Future Works	44
10.0 References	45

1.0 Introduction

1.1 Description of The Organization

AIU Shop, an enterprise established by Albukhary International University, is committed to the promotion of items sourced from rural areas and the support of social business products created by AIU students. Nevertheless, the existing manual system utilized for the management of client information, staff records, sales transactions, and inventories presents notable obstacles in terms of operational efficiency, data accuracy, and scalability. The primary objective of the proposed project is to create and execute a complete database management system that would effectively tackle the aforementioned difficulties, optimize operational processes, and support the realization of AIU Shop's social impact goals.

1.2 Description of The Current System and Its Problem

The current system mainly depends on manual record-keeping and different data sources, resulting in inefficiencies and mistakes. One of the primary issues that arise in this context are the challenges that need to be addressed. The presence of fragmented client data is a consequence of the absence of a centralized system, resulting in the dispersion of consumer information and impeding the provision of personalized services. The issue of ineffective personnel management is a significant concern. The practice of manually monitoring staff information and task progress leads to operational inefficiencies. Insufficient monitoring and analysis of sales performance. The lack of comprehensive understanding regarding sales patterns, client preferences, and transaction records is a significant obstacle to making well-informed decisions. The challenges associated with inventory management. The challenge of effectively managing precise stock levels and supplier information has a direct impact on the availability of products.

1.3 Motivation

One of the motivations of this database system is to increase AIU Shops' efficacy by substituting a contemporary database for the manual system. This change was brought about by the need to address problems with dispersed customer data, ineffective staff management, little sales intelligence, and difficult inventory management. Along with ensuring data quality and streamlining processes, the objective is to better assist AIU Shop's mission of promoting rural businesses and student-made social business products.

1.4 How will the organization gain from the integration of this system?

The integration of AIU Shop to e-commerce activities is one of the advantages of the AIU Shop database system. Consumers can use both physical and online platform ways to conduct transactions via providing delivery services and online transactions. The company can increase the ease of the purchasing process and reach a wider audience of customers. Furthermore, the system reduces administrative workloads by facilitating effective administration and arrangement of sales, inventory, staff, and customer data for AIU Shop.

1.5 Module

This AIU Shop's database system has 4 main modules inside including the customer module, employee module, sales module and inventory module.

- Customer Module

The Customer module is designed to facilitate the onboarding of new users by allowing them to create and manage their accounts. Within this database framework, the module captures essential customer details, including their name, identification number, address, email, phone number, username and password. User registration is mandatory for accessing the system, enabling them to explore the available products and make purchases. Upon selecting desired items, customers are prompted to specify their

preferred transaction method either through self-pickup or delivery. The delivery option incurs additional charges based on the customer's location. During the checkout process, customers have the flexibility to choose their preferred payment method, including cash, e-wallet, online banking, or debit card, ensuring a seamless and convenient transaction experience.

- Employee Module

The Employee module serves as the repository for crucial information about individuals employed by AIU Shop. Upon commencement of their tenure, the module records details such as the employee's name, identification number, phone number, username, password, address, and gender. Additionally, it encompasses the registration date, salary, qualifications, and employment status of each employee. Also this module empowers employees to access and manage information pertaining to customers, fellow employees, sales, and inventory based on their designated roles. Managers possess the authority to both view and update details related to customers, employees, inventory, and sales. Workers are granted access to customer, sales, and inventory information for viewing and updating. In contrast, assistant managers are limited to viewing and updating customer and inventory details exclusively. This structured approach ensures that specific positions have access to and can modify relevant information, contributing to the efficient operation of the pharmacy.

- Sales Module

The sales module allows the customer to purchase various products in the shop. The customer is required to log in to the account to be able to purchase a product. This module will display the products with available stock based on their categories such as PO Stich,Plastigo,Kerepek Mak Lang and Therapy Lab. To purchase a product, the customer is required to select the desired product, edit the required quantity, and add the product to their cart. All products sold are not non-returnable and non-refundable. After

the customer completes their payment, the number of products in stock will be reduced in the inventory. The sales record can be accessed by the employee to obtain information such as daily sales, customer counts, peak purchase time, and best-selling products.

- Inventory Module

The inventory module it will allow the employee to update any informations about AIU Shop inventory storage. This module will displays the management of inventory, encompassing several aspects such as the product catalog, stock levels, supplier information, and notifications for restocking, all aimed at ensuring the availability of products. The quantity of the products in stocks will be automatically deducted when a customers has already purchased the product in the AIU Shops. After the product finishes in the stock, the employee will replenish stock from the supplier and also the quantity of the product will be updated by the employee in the stock.

2.0 User Requirement and Business Rules

2.1 Customer Module

User Requirements	Business Rules
1. User Registration: Capture essential details for account creation (name, ID, address, contact, login credentials).	1. Mandatory Registration: Access to the system necessitates user registration.
2. Access Control: Mandate registration for system access; validate login credentials.	2. Secure Access: Implement encryption and security measures for user data and login credentials.

3. Transaction Preferences: Offer pickup/delivery options with varying charges based on location.	3. Transparent Charges: Clearly outline delivery fees based on customer location.
4. Payment Options: Enable multiple payment methods (cash, e-wallet, online banking, debit card).	4. Secure Transactions: Ensure encryption and secure protocols for all payment transactions.
5. Cart Management: Facilitate the addition, removal, and management of selected items before checkout.	5. User Privacy: Safeguard customer information and comply with data protection regulations.

2.2 Employee Module

User Requirements	Business Rules
1. Employee Information: Record employee details (personal, contact, role-specific information).	1. Role-Based Access: Grant access levels based on job roles to maintain data integrity.
2. Role-Based Access: Provide differentiated access rights for various roles (manager, worker, assistant manager).	2. Data Confidentiality: Enforce access control to sensitive employee data based on roles.
3. Data Management: Enable employees to handle customer, sales, and inventory data based on their designated roles.	3. Authorized Data Handling: Limit access to sensitive information based on employee roles.

2.3 Sales Module

User Requirements	Business Rules
1. Product Display: Categorize and display available items with relevant details.	1. Non-Returnable Policy: Clarify that all sold products are non-returnable and non-refundable.
2. User Authentication: Require user login for purchase transactions.	2. Real-time Inventory Update: Automatically reduce stock upon successful purchase.
3. Purchase Process: Enable product selection, quantity adjustment, and cart management.	3. Sales Analytics: Track sales trends for informed decision-making (e.g., popular items, peak sales times).
4. Transaction Finalization: Complete transactions and update inventory post-payment.	4. Accurate Inventory Management: Ensure immediate stock updates post-sale to maintain inventory accuracy.

2.4 Inventory Module

User Requirements	Business Rules
1. Inventory Management: Maintain product catalog, stock levels, supplier information, and restocking notifications.	1. Supplier Coordination: Facilitate restocking procedures and maintain communication with suppliers.
2. Stock Replenishment: Allow authorized employees to update stock levels after sales and replenish products from suppliers.	2. Automated Stock Updates: Ensure real-time stock adjustments for accurate inventory records.
3. Notification System: Generate alerts for low stock levels to prompt timely replenishment.	3. Efficient Inventory Handling: Prevent stock outs by timely stock updates and replenishment actions.

2.5 Cross Module User Requirements and Business Rules

User Requirements	Implementation Guidelines
1. Data Synchronization: Ensure seamless data flow across modules for consistency and accuracy.	- Implement APIs or middleware for real-time data synchronization. Schedule regular data synchronization checks.
2. Inter-Module Access Control: Enforce access limitations based on user roles across modules.	- Maintain a centralized access control system for uniform user permissions. Ensure consistent access protocols across modules.

3. Integration Efficiency: Establish smooth communication channels between modules for operational harmony.

- Use standardized data formats and protocols for seamless module interaction. Conduct thorough testing of inter-module communication channels.

3.0 Entity Relationship Modelling

All the entity relationship diagrams are drawn with Crow's foot notation

3.1 Customer Module

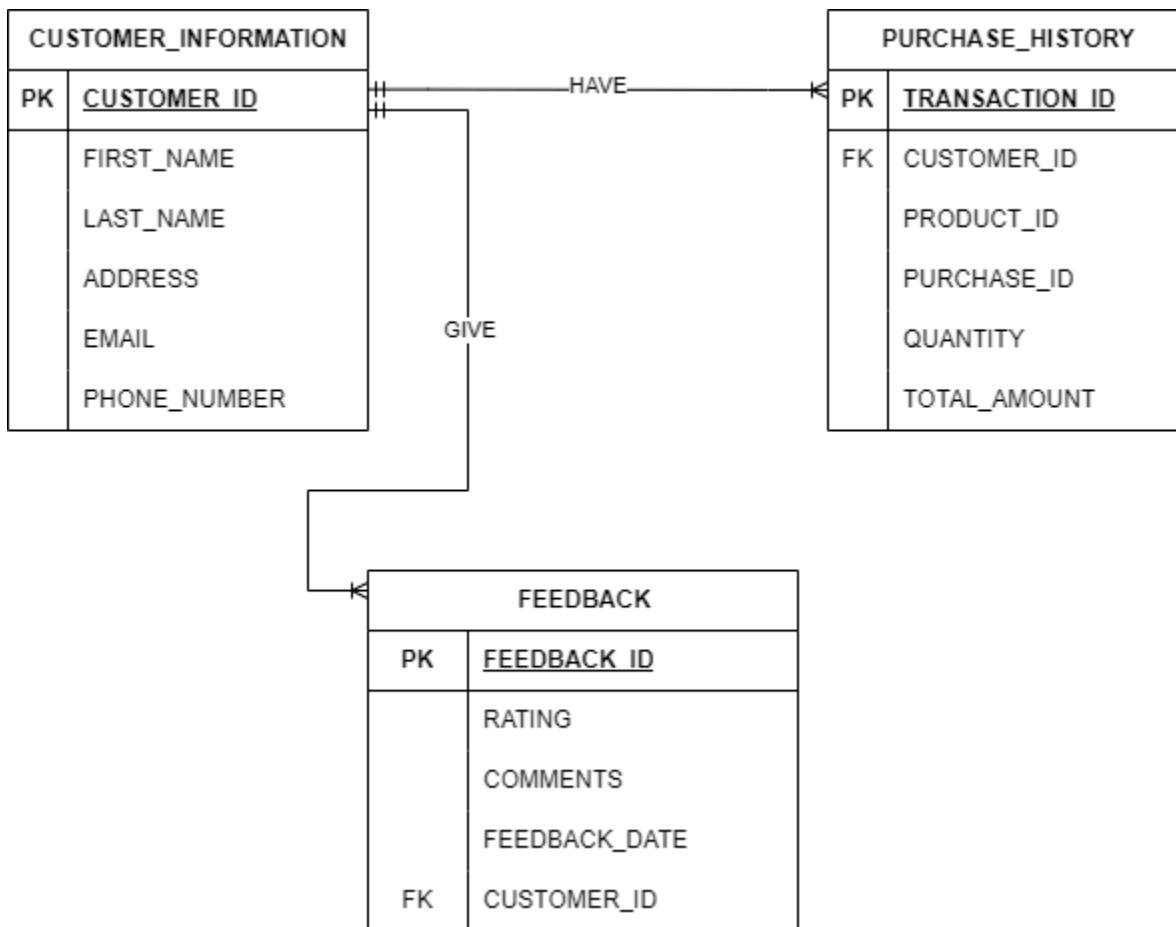


Figure 1 : ERD Diagram for Customer Module

Explanation:

Customer module contains 3 entities, namely CUSTOMER_INFORMATION, PURCHASE_HISTORY, and FEEDBACK.

Relationship:

- CUSTOMER_INFORMATION entity and PURCHASE_HISTORY has one-to-many relationship. PURCHASE_HISTORY entity is an optional entity to

CUSTOMER_INFORMATION entity while CUSTOMER_INFORMATION entity is a mandatory entity to PURCHASE_HISTORY entity.

- CUSTOMER_INFORMATION entity and FEEDBACK entity has one-to-many relationship between each other connection relation.

Key Constraints:

- The primary key of CUSTOMER_INFORMATION entity is CUSTOMER_ID.
- The primary key of PURCHASE_HISTORY entity is TRANSACTION_ID and inside the PURCHASE_HISTORY entity has a CUSTOMER_ID as foreign key inherited from CUSTOMER_INFORMATION entity.
- FEEDBACK entity also has CUSTOMER_ID as foreign key inherited from CUSTOMER_INFORMATION entity.

Clarification:

- In CUSTOMER_INFORMATION entity, CUSTOMER_ID is a primary key which is unique ID after they are registered into the system. The other attributes are FIRST_NAME, LAST_NAME, ADDRESS, EMAIL and PHONE_NUMBER. They represent the customer's first name, customer's last name, customer's address, customer's email, and customer's phone number.
- In PURCHASE_HISTORY entity, the primary key is TRANSACTION_ID which can be used to track and record purchases made by the customer. For the foreign key inherited from the CUSTOMER_INFORMATION is CUSTOMER_ID also to track the customer's transaction that has been made. The other attributes which are PRODUCT_ID that every item in the shop can know by the ID given. PURCHASE_ID attributes is the proof of the customer's made some purchase from this shop that can be recorded into the system. QUANTITY attributes it's to get the newest updated quantity item in the shop and to track every item in the shop that has been sold or not. TOTAL_AMOUNT attributes its to get the correct amount after made some purchasing item in the shop.
- In FEEDBACK entity, FEEDBACK_ID which is primary key is actually an ID generated by the system after the customer has successfully made some transactions. The CUSTOMER_ID its a primary key inherited from CUSTOMER_INFORMATION entity

that will track the customer feedback easily. The other attributes which are RATING is how the customer will rate the item they buy from the shops and how satisfied they are with the item they buy. COMMENT attributes the customers will comment about the product they buy and also the service they get. FEEDBACK_DATE attributes the date that the customers gave the feedback and it will insert to the system.

3.2 Employee Module

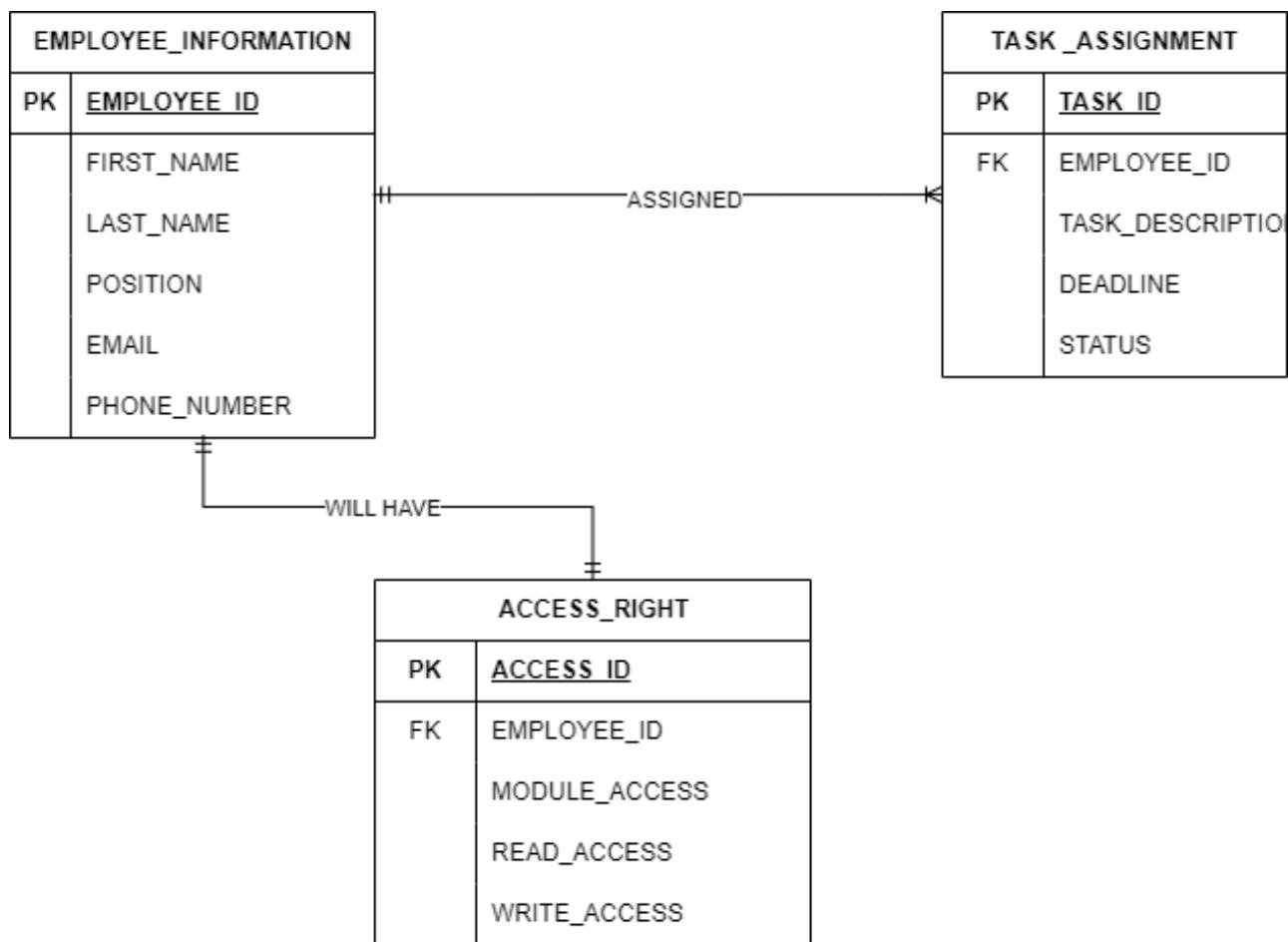


Figure 2: ERD Diagram for Employee Module

Explanation:

Employee module has 3 entities, which is EMPLOYEE_INFORMATION, TASK_ASSIGNMENT, and ACCESS_RIGHT.

Relationship:

- EMPLOYEE_INFORMATION entity and TASK_ASSIGNMENT has one-to-many relationship. TASK_ASSIGNMENT entity is an optional entity to EMPLOYEE_INFORMATION entity while EMPLOYEE_INFORMATION entity is a mandatory entity to TASK_ASSIGNMENT entity.
- ACCESS_RIGHT entity and EMPLOYEE_INFORMATION has one-to-one relationship.

Key Constraints:

- The primary key of the EMPLOYEE_INFORMATION entity is EMPLOYEE_ID which is unique ID after they are registered into the system.
- In TASK_ASSIGNMENT entity, TASK_ID is a primary key which is unique ID that already inserted in the system and inside the TASK_ASSIGNMENT entity has a foreign key that's EMPLOYEE_ID inherited from EMPLOYEE_INFORMATION entity.
- In ACCESS_RIGHT entity, ACCESS_ID is a primary key which is unique ID that already in the systems for the employee to access they id in the systems. In the ACCESS_RIGHT entity,EMPLOYEE_ID becomes the foreign key for this entity that inherited from EMPLOYEE_INFORMATION entity.

Clarification:

- In EMPLOYEE_INFORMATION, EMPLOYEE_ID is a primary key which is unique ID after they are registered into the system. The other attributes are FIRST_NAME, LAST_NAME, POSITION, EMAIL and PHONE_NUMBER. They represent the employee first name, employee last name, employee position, employee email and employee phone number.
- In TASK_ASSIGNMENT, TASK_ID is a primary key which is to track every work the employee do and dont do to make sure every task in the system. Attributes involved to store employee task information of each such as TASK_DESCRIPTION, DEADLINE,

and STATUS. TASK DESCRIPTION attributes that employee will know the task given and upcoming task. DEADLINE attributes that will make sure the employee has a deadline for every task they do. STATUS attributes to know the task status how its going.

- In ACCESS_RIGHT entity ,ACCESS_ID is the primary key that becomes the unique id for the employee get the right access to access the information. The other attributes which are MODULE_ACCESS, READ_ACCESS, and WRITE_ACCESS. MODULE_ACCESS attribute is to employee access the right module. READ_ACCESS attribute is to only allow the employee access the module only for read. WRITE_ACCESS attribute is for the employee read and write the information they want in the module.

3.3 Sales Module

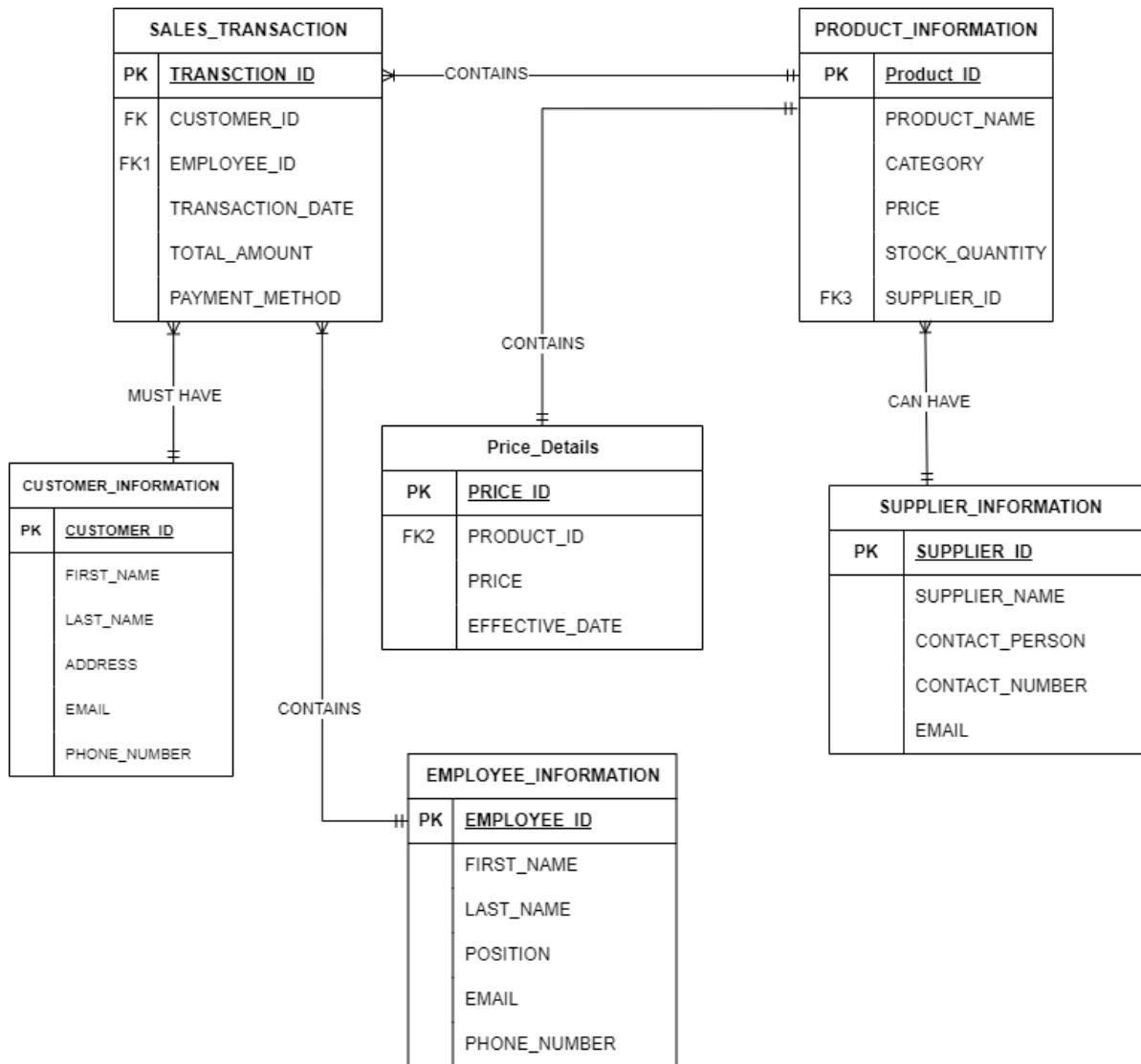


Figure 3: ERD Diagram for Sales Module

Explanation:

Customer module contain contains 3 main entities which is SALES_TRANSCATION, PRODUCT_INFORMATION and PRICE_DETAILS. Another 3 entities that are not mentioned

but exist in the diagram is entities that has relationship with this SALES MODULE but comes from another MODULE.

Relationship:

- SALES_transaction entity and product_information entity has one-to-many .
- PRODUCT_INFORMATION entity and SUPPLIER_INFORMATION has one-to-many relationship.
- PRICE_DETAILS entity and PRODUCT INFORMATION entity has one-to-one relationship.
- For SUPPLIER_INFORMATION, EMPLOYEE_INFORMATION and CUSTOMER_INFORMATION. This is just entities that we put just for make SALES MODULE complete because there is some relationship with others module.

Key Constraints:

- The primary key of SALES_TRANSACTION entity is TRANSACTION_ID and inside the SALES_TRANSACTION entity has foreign key which is CUSTOMER_ID as primary key inherited from CUSTOMER_INFORMATION entity and EMPLOYEE_ID as primary key inherited from EMPLOYEE_INFORMATION entity.
- The primary key of PRODUCT_INFORMATION entity is PRODUCT_ID and inside the PRODUCT_INFORMATION entity has foreign key which is SUPPLIER_ID as primary key inherited from SUPPLIER_INFORMATION entity.
- The primary key of PRICE_DETAILS entity is PRICE_ID and inside the PRICE_DETAILS entity has foreign key which is PRODUCT_ID as primary key inherited from PRODUCT_INFORMATION entity

Clarification:

- In SALES_TRANSACTION entity, TRANSACTION_ID is a primary key which is unique ID after they are registered into the system. The other attributes are CUSTOMER_ID, EMPLOYEE_ID, TRANSACTION_DATE, TOTAL_AMOUNT and

PAYMENT_METHOD. Those attributes are for transaction id, customer id, employee id, transaction date, total amount and payment method such as via cash, card, online.

- In PRODUCT_INFORMATION entity, the primary key is PRODUCT_ID which is unique ID after they are registered into the system. The other attributes are PRODUCT_NAME, CATEGORY, PRICE, STOCK_QUANTITY and SUPPLIER_ID. Those attributes are for product name, category, price, stock quantity and supplier id. Inside the PRICE_DETAILS entity has foreign key which is PRODUCT_ID as primary key inherited from PRODUCT_INFORMATION entity
- In PRICE_DETAILS entity, the primary key is PRICE_ID which is unique ID after they are registered into the system. The other attributes are PRODUCT_ID, PRICE and EFFECTIVE_DATE. Those attributes are for product id, pricelist and effective date. Inside the PRICE_DETAILS entity has foreign key which is PRODUCT_ID as primary key inherited from PRODUCT_INFORMATION entity.

3.4 Inventory Module

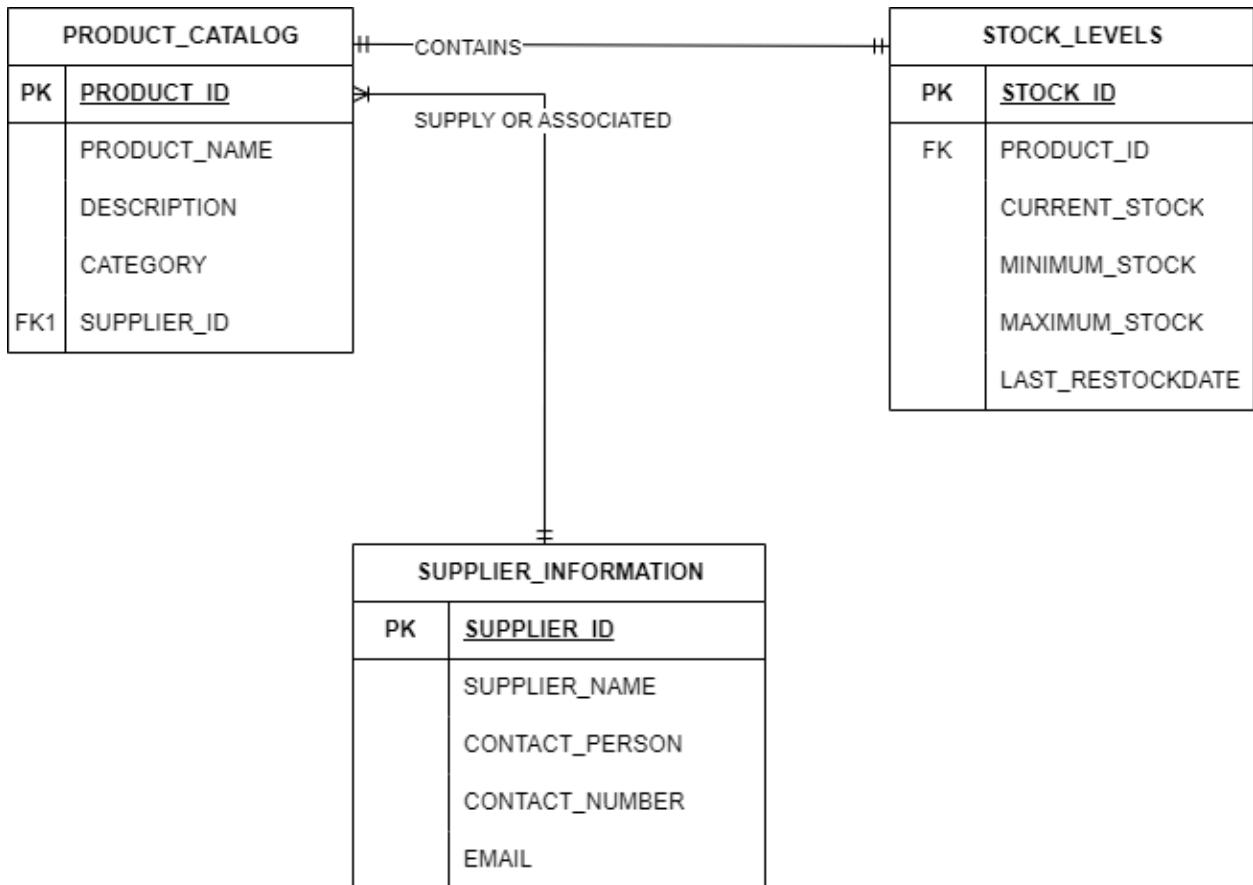


Figure 4: ERD Diagram for Inventory Module

Explanation:

Inventory module contains 3 entities which includes PRODUCT_CATALOG, STOCK_LEVELS, and SUPPLIER_INFORMATION.

Relationship:

- PRODUCT_CATALOG entity and STOCK_LEVELS entity has one-to-one relationship.
- SUPPLIER_INFORMATION entity and PRODUCT_CATALOG entity has one-to-many relationship. PRODUCT_CATALOG entity is an optional entity to SUPPLIER_INFORMATION entity while SUPPLIER_INFORMATION entity is a mandatory entity to PRODUCT_CATALOG entity.

Key Constraints:

- The primary key of PRODUCT_CATALOG entity is PRODUCT_ID that already become the unique id that has been included inside the system.. It contains foreign key which are SUPPLIER_ID.
- In the STOCK_LEVELS entity , STOCK_ID is a primary key for STOCK_LEVELS entity that become unique in the entity.Inside the STOCK_LEVELS entity has a foreign key which is PRODUCT_ID.
- SUPPLIER_INFORMATION entity had a primary key which is SUPPLIER_ID also become the unique to make sure every supplier has the right id in the system.Also to determined SUPPLIER_NAME, CONTACT_PERSON and CONTACT_NUMBER and EMAIL.

Clarification:

- In PRODUT_CATALOG entity, PRODUCT_ID is a unique ID generated when a product is recorded into the system. The other attributes such as PRODUCT_NAME, DESCRIPTON, CATEGORY and SUPPLIER_ID .PORUCT_NAME attributes is to name all the items included in the stock shops. DESCRIPTON attributes is to know all about description of every items. CATEGORY attributes is to put evertime in their category which category should be stored.
- In STOCK_LEVELS entity, STOCK_ID is the primary key of this entity for this system to get know and track every items and stock in the shops. The other attributes such as CURRENT_STOCK, MINIMUM_STOCK, MAXIMUM_STOCK and

LAST_RESTOCKDATE is to record the product current stock,minimum stock, maximum stock ,and record of the restock date.

- Inside SUPPLIER_INFORMATION, the primary key of this entity is SUPPLIER_ID and the attributes of this entity is SUPPLIER_NAME, CONTACT_PERSON, CONTACT_NUMBER and EMAIL. All of this attributes is to

3.5 Combine ERD AIU SHOPS

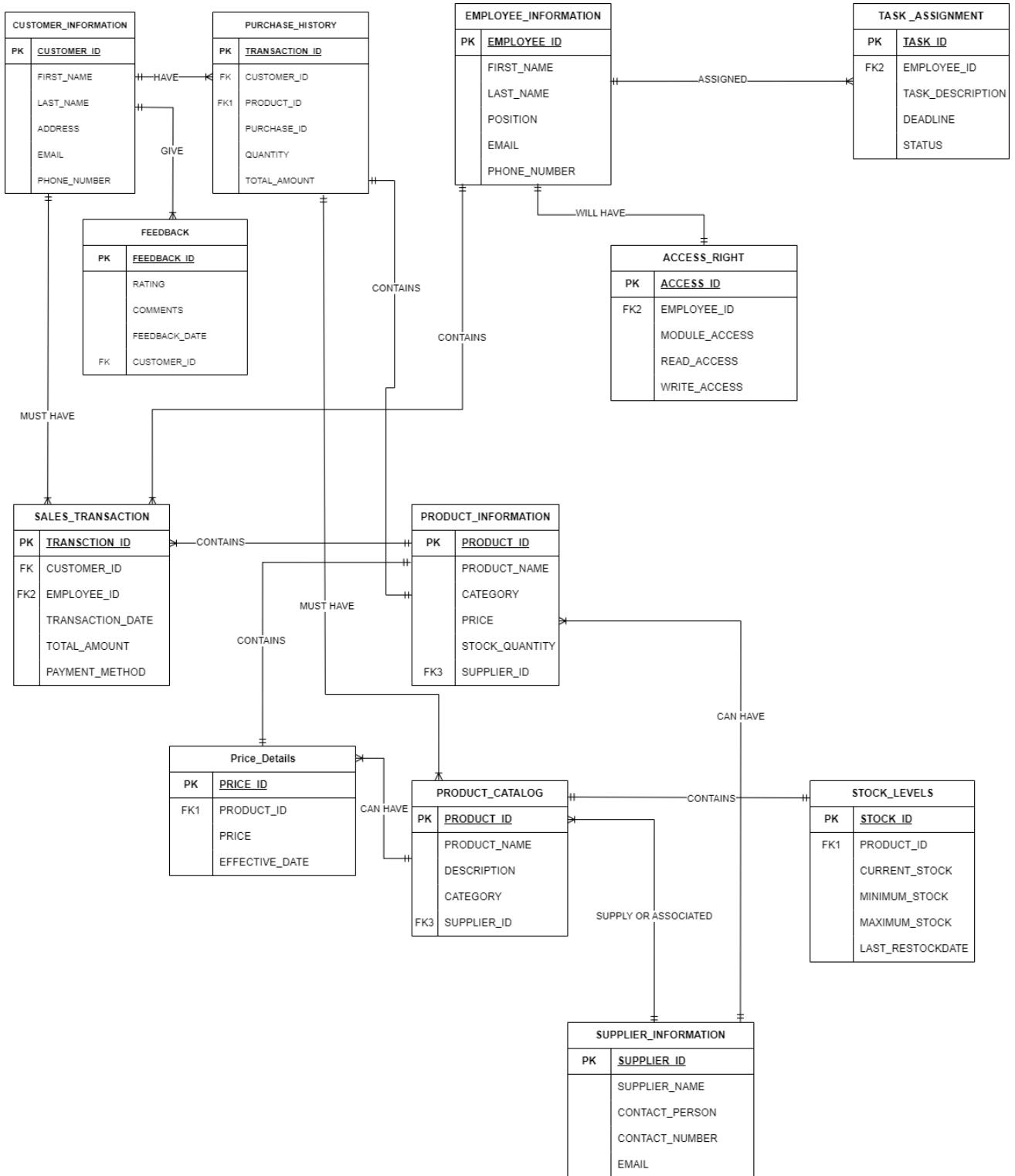


Figure 5: ERD Diagram for entire AIU SHOPS DATABASE

Explanation :

This ERD is the combination of all the four modules. This ERD contains 12 entities and 14 relationships in total. After the combination of all 4 modules, there are 3 relationships that overlap with each module.

Relationships :

- CUSTOMER_INFORMATION entity and PURCHASE_HISTORY has one-to-many relationship. PURCHASE_HISTORY entity is an optional entity to CUSTOMER_INFORMATION entity while CUSTOMER_INFORMATION entity is a mandatory entity to PURCHASE_HISTORY entity.
- CUSTOMER_INFORMATION entity and FEEDBACK entity has oen-to-many relationship between each other connection relation.
- CUSTOMER_INFORMATION entity and SALES_TRANSACTION entity has one-to-many relationship between each other connection relation.
- EMPLOYEE_INFORMATION entity and TASK_ASSIGNMENT has one-to-many relationship. TASK_ASSIGNMENT entity is an optional entity to EMPLOYEE_INFORMATION entity while EMPLOYEE_INFORMATION entity is a mandatory entity to TASK_ASSIGNMENT entity.
- EMPLOYEE_INFORMATION entity to SALES_TRANSACTION has one-to-many relationship.
- ACCESS_RIGHT entity and EMPLOYEE_INFORMATION has one-to-one relationship.
- SALES_transaction entity and PRODUCT_INFORMATION entity has one-to-many .
- PRODUCT_INFORMATION entity and SUPPLIER_INFORMATION has one-to-many relationship.
- PRODUCT_INFORMATION entity and PRICE_DETAILS has one-to-one relationship.
- PRODUCT_INFORMATION entity and SALES_TRANSACTION has one-to-many relationship.
- PRODUCT_INFORMATION entity and PURCHASE_HISTORY has one-to-one relationship.

- PRODUCT_CATALOG entity and STOCK_LEVELS entity has one-to-one relationship.
- PRODUCT_CATALOG entity and PRICE_DETAILS entity has one-to-many relationship.
- SUPPLIER_INFORMATION entity and PRODUCT_CATALOG entity has one-to-many relationship. PRODUCT_CATALOG entity is an optional entity to SUPPLIER_INFORMATION entity while SUPPLIER_INFORMATION entity is a mandatory entity to PRODUCT_CATALOG entity.

4.0 Normalization

Normalization is a database design technique that avoids redundancy and ensures data integrity by organizing data into smaller, more focused tables and creating linkages between them. Its goal is to optimize storage and retrieval while avoiding errors including insertion, deletion, and update anomalies. Normalization improves data reliability by eliminating redundant information and guaranteeing uniformity. It takes numerous forms, including 1NF, 2NF, and 3NF, to address concerns like partial and transitive dependencies. While higher forms, such as BCNF, provide more durable structures, the level of normalization should be calibrated to meet the specific needs of the data.

4.1 Customer Module

All tables are in 3NF which do not contain partial dependency and transitive dependency. Below shows the dependency diagrams of the tables.

NOTE: Coloured box are primary keys.

Table : CUSTOMER_INFORMATION (3NF)

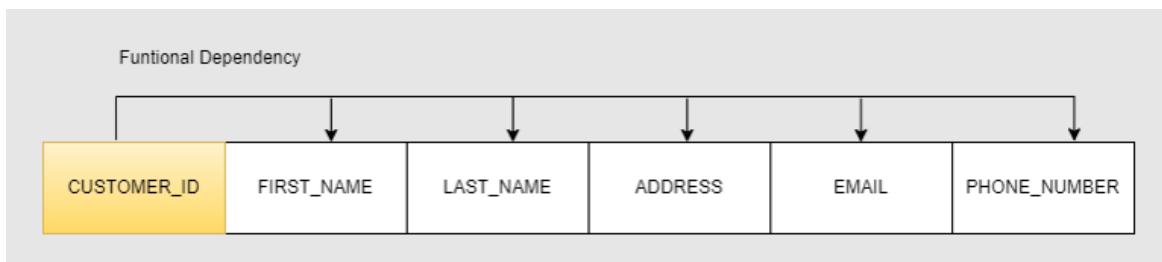


Table : PURCHASE_HISTORY (3NF)

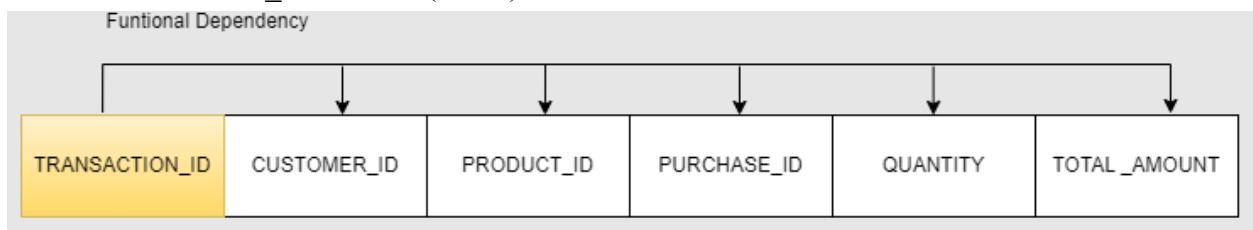
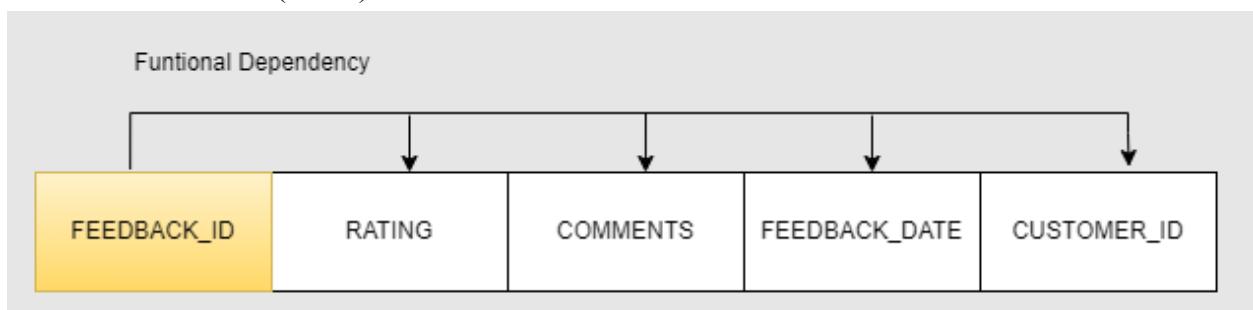


Table : FEEDBACK (3NF)



4.2 Employee Module

Tables named EMPLOYEE_INFORMATION, TASK_ASSIGNMENT and ACCESS_RIGHT in the employee module are in 3NF because there are no partial and transitive dependencies.

Table : EMPLOYEE_INFORMATION (3NF)

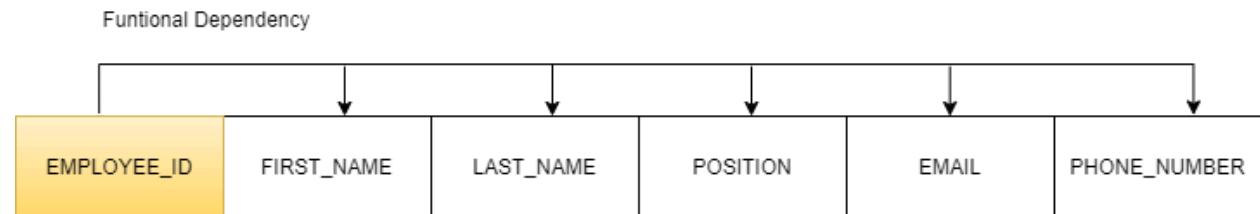


Table : TASK_ASSIGNMENT (3NF)

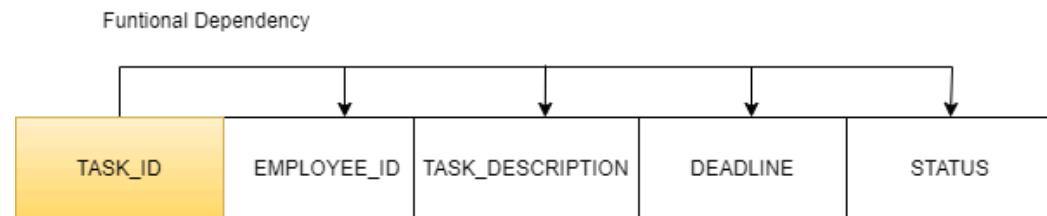
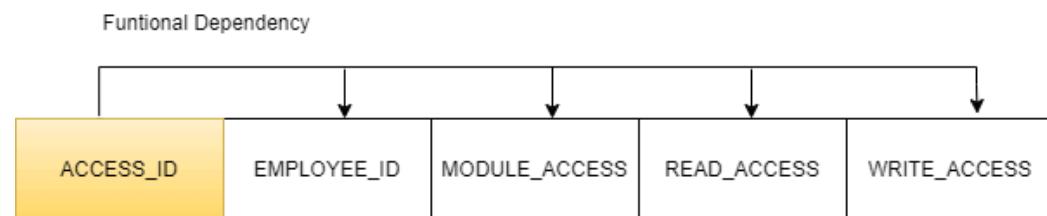


Table : ACCESS_RIGHT (3NF)



4.3 Sales Module

All tables are in 3NF which do not contain partial dependency and transitive dependency. Below shows the dependency diagrams of the tables.

NOTE: Coloured box are primary keys.

Table : CUSTOMER_INFORMATION (3NF)

Functional Dependency

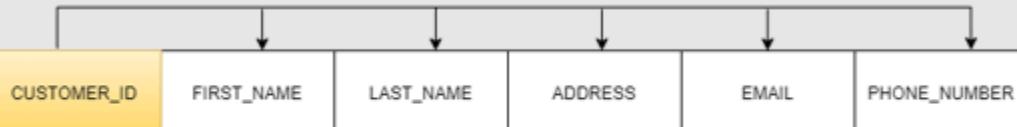


Table : EMPLOYEE_INFORMATION (3NF)

Functional Dependency

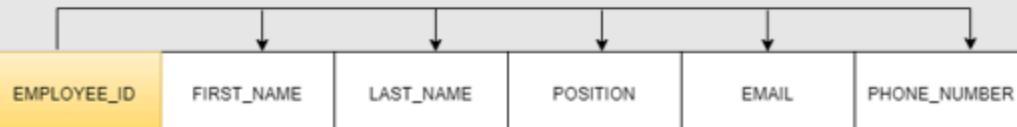


Table : PRODUCT_INFORMATION (3NF)

Functional Dependency



Table : SUPPLIER_INFORMATION (3NF)

Functional Dependency

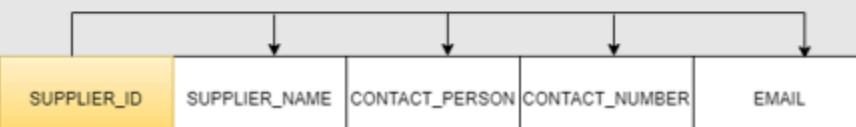
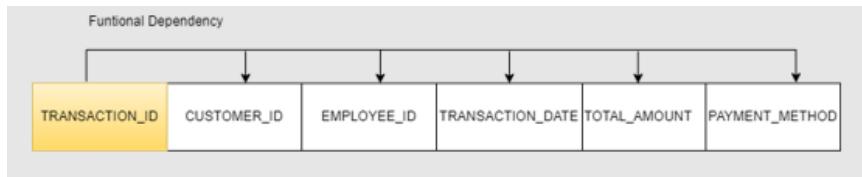


Table : PRICE_DETAILS (3NF)

Functional Dependency



Table : SALES_TRANSACTIONS (3NF)



4.4 Inventory Module

All three tables in the inventory module are in 3NF normalizations because there are no partial dependencies and transitive dependencies. Below this shows that the dependencies diagrams of the tables. For this module partial dependencies and transitive dependencies are absent.

Table : PRODUCT_CATALOG (3NF)

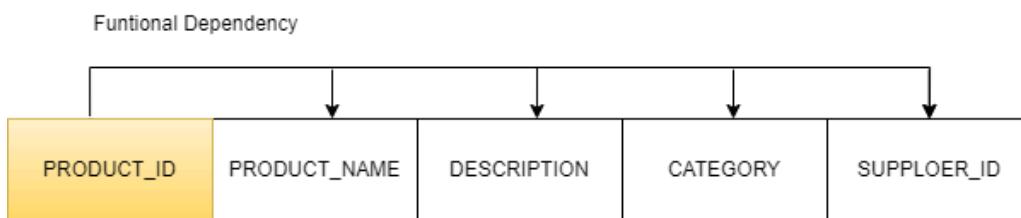


Table : STOCK_LEVELS (3NF)

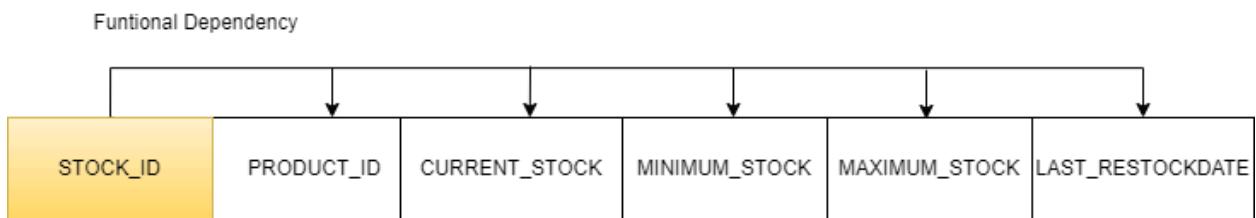
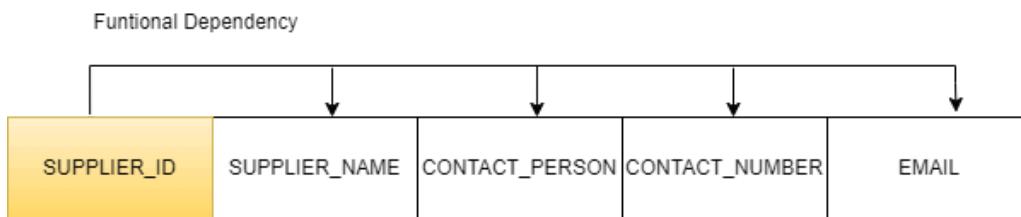


Table : SUPPLIER_INFORMATION (3NF)



5.0 Data dictionary

5.1 Customer Module

aui_shop	customer_information	TBL	1 PK	customer_id	int(10)	NOT NULL
aui_shop	customer_information	TBL	2	first_name	varchar(50)	NOT NULL
aui_shop	customer_information	TBL	3	last_name	varchar(50)	NOT NULL
aui_shop	customer_information	TBL	4	address	varchar(255)	NULL
aui_shop	customer_information	TBL	5	email	varchar(50)	NULL
aui_shop	customer_information	TBL	6	phone_number	varchar(20)	NULL
aui_shop	feedback	TBL	1 PK	feedback_id	int(10)	NOT NULL
aui_shop	feedback	TBL	2	rating	int(10)	NOT NULL
aui_shop	feedback	TBL	3	comments	varchar(255)	NULL
aui_shop	feedback	TBL	4	feedback_date	date(3)	NOT NULL
aui_shop	feedback	TBL	5 FK	customer_id	int(10)	NULL
aui_shop	purchase_history	TBL	1 PK	transaction_id	int(10)	NOT NULL
aui_shop	purchase_history	TBL	2 FK	customer_id	int(10)	NULL
aui_shop	purchase_history	TBL	3 FK	product_id	int(10)	NULL
aui_shop	purchase_history	TBL	4	purchase_id	int(10)	NULL
aui_shop	purchase_history	TBL	5	quantity	int(10)	NOT NULL
aui_shop	purchase_history	TBL	6	total_amount	decimal(10,2)	NOT NULL

5.2 Employee Module

aui_shop	employee_information	TBL	1 PK	employee_id	int(10)	NOT NULL
aui_shop	employee_information	TBL	2	first_name	varchar(50)	NOT NULL
aui_shop	employee_information	TBL	3	last_name	varchar(50)	NOT NULL
aui_shop	employee_information	TBL	4	position	varchar(50)	NULL
aui_shop	employee_information	TBL	5	email	varchar(50)	NULL
aui_shop	employee_information	TBL	6	phone_number	varchar(20)	NULL
aui_shop	task_assignment	TBL	1 PK	task_id	int(10)	NOT NULL
aui_shop	task_assignment	TBL	2 FK	employee_id	int(10)	NULL
aui_shop	task_assignment	TBL	3	task_description	varchar(255)	NOT NULL
aui_shop	task_assignment	TBL	4	deadline	date(3)	NULL
aui_shop	task_assignment	TBL	5	status	varchar(20)	NULL
aui_shop	access_right	TBL	1 PK	access_id	int(10)	NOT NULL
aui_shop	access_right	TBL	2 FK	employee_id	int(10)	NULL
aui_shop	access_right	TBL	3	module_access	varchar(50)	NOT NULL
aui_shop	access_right	TBL	4	read_access	tinyint(3)	NOT NULL
aui_shop	access_right	TBL	5	write_access	tinyint(3)	NOT NULL

5.3 Sales Module

aiu_shop	sales_transaction	TBL	1	PK	transaction_id	int(10)	NOT NULL
aiu_shop	sales_transaction	TBL	2	FK	customer_id	int(10)	NULL
aiu_shop	sales_transaction	TBL	3	FK	employee_id	int(10)	NULL
aiu_shop	sales_transaction	TBL	4		transaction_date	date(3)	NOT NULL
aiu_shop	sales_transaction	TBL	5		total_amount	decimal(10,2)	NOT NULL
aiu_shop	sales_transaction	TBL	6		payment_method	varchar(20)	NULL
aiu_shop	product_information	TBL	1	PK	product_id	int(10)	NOT NULL
aiu_shop	product_information	TBL	2		product_name	varchar(50)	NOT NULL
aiu_shop	product_information	TBL	3		category	varchar(50)	NULL
aiu_shop	product_information	TBL	4		price	decimal(10,2)	NOT NULL
aiu_shop	product_information	TBL	5		stock_quantity	int(10)	NOT NULL
aiu_shop	product_information	TBL	6	FK	supplier_id	int(10)	NULL
aiu_shop	price_details	TBL	1	PK	price_id	int(10)	NOT NULL
aiu_shop	price_details	TBL	2	FK	product_id	int(10)	NULL
aiu_shop	price_details	TBL	3		price	decimal(10,2)	NOT NULL
aiu_shop	price_details	TBL	4		effective_date	date(3)	NOT NULL
aiu_shop	customer_information	TBL	1	PK	customer_id	int(10)	NOT NULL
aiu_shop	customer_information	TBL	2		first_name	varchar(50)	NOT NULL
aiu_shop	customer_information	TBL	3		last_name	varchar(50)	NOT NULL
aiu_shop	customer_information	TBL	4		address	varchar(255)	NULL
aiu_shop	customer_information	TBL	5		email	varchar(50)	NULL
aiu_shop	customer_information	TBL	6		phone_number	varchar(20)	NULL
aiu_shop	supplier_information	TBL	1	PK	supplier_id	int(10)	NOT NULL
aiu_shop	supplier_information	TBL	2		supplier_name	varchar(50)	NOT NULL
aiu_shop	supplier_information	TBL	3		contact_person	varchar(50)	NULL
aiu_shop	supplier_information	TBL	4		contact_number	varchar(20)	NULL
aiu_shop	supplier_information	TBL	5		email	varchar(50)	NULL
aiu_shop	customer_information	TBL	1	PK	customer_id	int(10)	NOT NULL
aiu_shop	customer_information	TBL	2		first_name	varchar(50)	NOT NULL
aiu_shop	customer_information	TBL	3		last_name	varchar(50)	NOT NULL
aiu_shop	customer_information	TBL	4		address	varchar(255)	NULL
aiu_shop	customer_information	TBL	5		email	varchar(50)	NULL
aiu_shop	customer_information	TBL	6		phone_number	varchar(20)	NULL
aiu_shop	employee_information	TBL	1	PK	employee_id	int(10)	NOT NULL
aiu_shop	employee_information	TBL	2		first_name	varchar(50)	NOT NULL
aiu_shop	employee_information	TBL	3		last_name	varchar(50)	NOT NULL
aiu_shop	employee_information	TBL	4		position	varchar(50)	NULL
aiu_shop	employee_information	TBL	5		email	varchar(50)	NULL
aiu_shop	employee_information	TBL	6		phone_number	varchar(20)	NULL

5.4 Inventory Module

aiu_shop	product_catalog	TBL	1	PK	product_id	int(10)	NOT NULL
aiu_shop	product_catalog	TBL	2		product_name	varchar(50)	NOT NULL
aiu_shop	product_catalog	TBL	3		description	varchar(255)	NULL
aiu_shop	product_catalog	TBL	4		category	varchar(50)	NULL
aiu_shop	product_catalog	TBL	5	FK	supplier_id	int(10)	NULL
aiu_shop	stock_levels	TBL	1	PK	stock_id	int(10)	NOT NULL
aiu_shop	stock_levels	TBL	2	FK	product_id	int(10)	NULL
aiu_shop	stock_levels	TBL	3		current_stock	int(10)	NOT NULL
aiu_shop	stock_levels	TBL	4		minimum_stock	int(10)	NOT NULL
aiu_shop	stock_levels	TBL	5		maximum_stock	int(10)	NOT NULL
aiu_shop	stock_levels	TBL	6		last_restockdate	date(3)	NULL
aiu_shop	supplier_information	TBL	1	PK	supplier_id	int(10)	NOT NULL
aiu_shop	supplier_information	TBL	2		supplier_name	varchar(50)	NOT NULL
aiu_shop	supplier_information	TBL	3		contact_person	varchar(50)	NULL
aiu_shop	supplier_information	TBL	4		contact_number	varchar(20)	NULL
aiu_shop	supplier_information	TBL	5		email	varchar(50)	NULL

6.0 Database Implementation

```
CREATE SCHEMA IF NOT EXISTS `aiu_shop` DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci ;
USE `aiu_shop` ;
```

```
-- Table `aiu_shop`.`employee_information`
```

```
CREATE TABLE IF NOT EXISTS `aiu_shop`.`employee_information` (
  `employee_id` INT NOT NULL,
  `first_name` VARCHAR(50) NOT NULL,
  `last_name` VARCHAR(50) NOT NULL,
  `position` VARCHAR(50) NULL DEFAULT NULL,
  `email` VARCHAR(50) NULL DEFAULT NULL,
  `phone_number` VARCHAR(20) NULL DEFAULT NULL,
  PRIMARY KEY (`employee_id`)
)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```
-- Table `aiu_shop`.`access_right`
```

```
CREATE TABLE IF NOT EXISTS `aiu_shop`.`access_right` (
  `access_id` INT NOT NULL,
```

```

`employee_id` INT NULL DEFAULT NULL,
`module_access` VARCHAR(50) NOT NULL,
`read_access` TINYINT(1) NOT NULL,
`write_access` TINYINT(1) NOT NULL,
PRIMARY KEY (`access_id`),
INDEX `employee_id`(`employee_id` ASC) VISIBLE,
CONSTRAINT `access_right_ibfk_1`
FOREIGN KEY (`employee_id`)
REFERENCES `aiu_shop`.`employee_information`(`employee_id`)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

```

-- -----
-- Table `aiu_shop`.`customer_information`

CREATE TABLE IF NOT EXISTS `aiu_shop`.`customer_information` (
`customer_id` INT NOT NULL,
`first_name` VARCHAR(50) NOT NULL,
`last_name` VARCHAR(50) NOT NULL,
`address` VARCHAR(255) NULL DEFAULT NULL,
`email` VARCHAR(50) NULL DEFAULT NULL,
`phone_number` VARCHAR(20) NULL DEFAULT NULL,
PRIMARY KEY (`customer_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

```

-- -----
-- Table `aiu_shop`.`feedback`

CREATE TABLE IF NOT EXISTS `aiu_shop`.`feedback` (
`feedback_id` INT NOT NULL,
`rating` INT NOT NULL,
`comments` VARCHAR(255) NULL DEFAULT NULL,
`feedback_date` DATE NOT NULL,
`customer_id` INT NULL DEFAULT NULL,
PRIMARY KEY (`feedback_id`),
```

```
INDEX `customer_id` (`customer_id` ASC) VISIBLE,
CONSTRAINT `feedback_ibfk_1`
  FOREIGN KEY (`customer_id`)
  REFERENCES `aiu_shop`.`customer_information` (`customer_id`)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```
-- -----  
-- Table `aiu_shop`.`supplier_information`  
-----
```

```
CREATE TABLE IF NOT EXISTS `aiu_shop`.`supplier_information` (
  `supplier_id` INT NOT NULL,
  `supplier_name` VARCHAR(50) NOT NULL,
  `contact_person` VARCHAR(50) NULL DEFAULT NULL,
  `contact_number` VARCHAR(20) NULL DEFAULT NULL,
  `email` VARCHAR(50) NULL DEFAULT NULL,
  PRIMARY KEY (`supplier_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```
-- -----  
-- Table `aiu_shop`.`product_catalog`  
-----
```

```
CREATE TABLE IF NOT EXISTS `aiu_shop`.`product_catalog` (
  `product_id` INT NOT NULL,
  `product_name` VARCHAR(50) NOT NULL,
  `description` VARCHAR(255) NULL DEFAULT NULL,
  `category` VARCHAR(50) NULL DEFAULT NULL,
  `supplier_id` INT NULL DEFAULT NULL,
  PRIMARY KEY (`product_id`),
  INDEX `supplier_id` (`supplier_id` ASC) VISIBLE,
  CONSTRAINT `product_catalog_ibfk_1`
    FOREIGN KEY (`supplier_id`)
    REFERENCES `aiu_shop`.`supplier_information` (`supplier_id`)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
```

```

COLLATE = utf8mb4_0900_ai_ci;

-- -----
-- Table `aiu_shop`.`price_details`
-- -----
CREATE TABLE IF NOT EXISTS `aiu_shop`.`price_details` (
  `price_id` INT NOT NULL,
  `product_id` INT NULL DEFAULT NULL,
  `price` DECIMAL(10,2) NOT NULL,
  `effective_date` DATE NOT NULL,
  PRIMARY KEY (`price_id`),
  INDEX `product_id` (`product_id` ASC) VISIBLE,
  CONSTRAINT `price_details_ibfk_1`
    FOREIGN KEY (`product_id`)
    REFERENCES `aiu_shop`.`product_catalog` (`product_id`)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-- -----
-- Table `aiu_shop`.`product_information`
-- -----
CREATE TABLE IF NOT EXISTS `aiu_shop`.`product_information` (
  `product_id` INT NOT NULL,
  `product_name` VARCHAR(50) NOT NULL,
  `category` VARCHAR(50) NULL DEFAULT NULL,
  `price` DECIMAL(10,2) NOT NULL,
  `stock_quantity` INT NOT NULL,
  `supplier_id` INT NULL DEFAULT NULL,
  PRIMARY KEY (`product_id`),
  INDEX `supplier_id` (`supplier_id` ASC) VISIBLE,
  CONSTRAINT `product_information_ibfk_1`
    FOREIGN KEY (`supplier_id`)
    REFERENCES `aiu_shop`.`supplier_information` (`supplier_id`)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

```

-- -----
-- Table `aiu_shop`.`purchase_history`

CREATE TABLE IF NOT EXISTS `aiu_shop`.`purchase_history` (
  `transaction_id` INT NOT NULL,
  `customer_id` INT NULL DEFAULT NULL,
  `product_id` INT NULL DEFAULT NULL,
  `purchase_id` INT NULL DEFAULT NULL,
  `quantity` INT NOT NULL,
  `total_amount` DECIMAL(10,2) NOT NULL,
  PRIMARY KEY (`transaction_id`),
  INDEX `customer_id` (`customer_id` ASC) VISIBLE,
  INDEX `product_id` (`product_id` ASC) VISIBLE,
  CONSTRAINT `purchase_history_ibfk_1`
    FOREIGN KEY (`customer_id`)
    REFERENCES `aiu_shop`.`customer_information` (`customer_id`),
  CONSTRAINT `purchase_history_ibfk_2`
    FOREIGN KEY (`product_id`)
    REFERENCES `aiu_shop`.`product_catalog` (`product_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

```

-- -----
-- Table `aiu_shop`.`sales_transaction`

CREATE TABLE IF NOT EXISTS `aiu_shop`.`sales_transaction` (
  `transaction_id` INT NOT NULL,
  `customer_id` INT NULL DEFAULT NULL,
  `employee_id` INT NULL DEFAULT NULL,
  `transaction_date` DATE NOT NULL,
  `total_amount` DECIMAL(10,2) NOT NULL,
  `payment_method` VARCHAR(20) NULL DEFAULT NULL,
  PRIMARY KEY (`transaction_id`),
  INDEX `customer_id` (`customer_id` ASC) VISIBLE,
  INDEX `employee_id` (`employee_id` ASC) VISIBLE,
  CONSTRAINT `sales_transaction_ibfk_1`
    FOREIGN KEY (`customer_id`)

```

```

REFERENCES `aiu_shop`.`customer_information`(`customer_id`),
CONSTRAINT `sales_transaction_ibfk_2`
FOREIGN KEY(`employee_id`)
REFERENCES `aiu_shop`.`employee_information`(`employee_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

-- Table `aiu_shop`.`stock_levels`

```

CREATE TABLE IF NOT EXISTS `aiu_shop`.`stock_levels` (
`stock_id` INT NOT NULL,
`product_id` INT NULL DEFAULT NULL,
`current_stock` INT NOT NULL,
`minimum_stock` INT NOT NULL,
`maximum_stock` INT NOT NULL,
`last_restockdate` DATE NULL DEFAULT NULL,
PRIMARY KEY (`stock_id`),
INDEX `product_id`(`product_id` ASC) VISIBLE,
CONSTRAINT `stock_levels_ibfk_1`
FOREIGN KEY(`product_id`)
REFERENCES `aiu_shop`.`product_catalog`(`product_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

-- Table `aiu_shop`.`task_assignment`

```

CREATE TABLE IF NOT EXISTS `aiu_shop`.`task_assignment` (
`task_id` INT NOT NULL,
`employee_id` INT NULL DEFAULT NULL,
`task_description` VARCHAR(255) NOT NULL,
`deadline` DATE NULL DEFAULT NULL,
`status` VARCHAR(20) NULL DEFAULT NULL,
PRIMARY KEY (`task_id`),
INDEX `employee_id`(`employee_id` ASC) VISIBLE,

```

```
CONSTRAINT `task_assignment_ibfk_1`  
FOREIGN KEY (`employee_id`)  
REFERENCES `aiu_shop`.`employee_information`(`employee_id`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_0900_ai_ci;
```

```
-- -----  
-- Table `aiu_shop`.`users`  
-- -----  
CREATE TABLE IF NOT EXISTS `aiu_shop`.`users` (  
`id` INT NOT NULL AUTO_INCREMENT,  
`username` VARCHAR(50) NOT NULL,  
`password` VARCHAR(255) NOT NULL,  
PRIMARY KEY (`id`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 5  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_0900_ai_ci;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

7.0 Front-end System Design and Implementation.

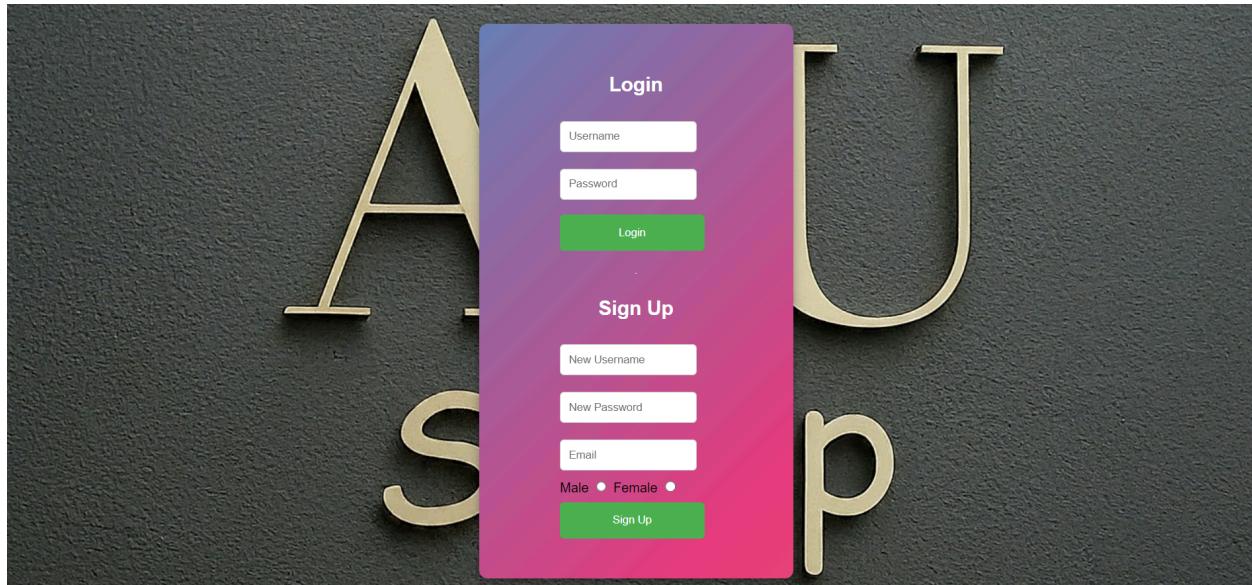


Figure 1: Login Page

The figure above is the login page of our AIU Shop Database. New users are required to sign up first and after that they can login with their usernames and password. Username and password are case-sensitive.

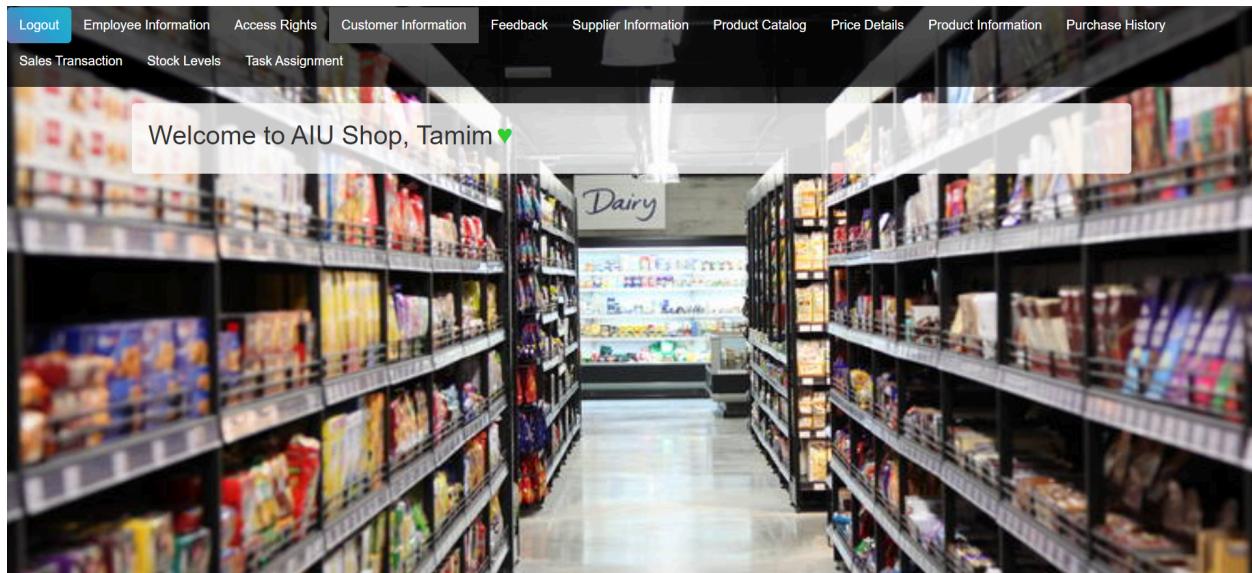


Figure 2: Home Page

Employee ID:

First Name:

Last Name:

Position:

Email:

Phone Number:

Add Employee

Figure 3: Edit Information

Existing Employees

Employee_id	First_name	Last_name	Position	Email	Phone_number	Actions
1	ali	ali	manager	tamim.lodin@student.aiu.edu.my	09099	Edit Delete
10	tamim	ahamd	HR	adsffas@sf	3563	Edit Delete
41	az	aq	staff	aksdfjl@sfja.com	76876	Edit Delete
3453	sdfg	sdffg	sdgg	sgsf	54654	Edit Delete

Figure 4: Existing Employees

Manage Access Right

Add New Access Right

Existing Access Rights

Access id	Employee id	Module access	Read access	Write access	Actions
-----------	-------------	---------------	-------------	--------------	---------

Figure 5: Manage Access Right

The screenshot shows a table with one row of customer information:

Customer ID	First Name	Last Name	Address	Email	Phone Number	Actions
4534	sgd	sgfg	sgdf	gsdf@afs	57656767	Edit Delete

Figure 6: Adding Customer Information

The screenshot shows a table with one row of feedback information:

ID	Rating	Comments	Date	Customer ID	Actions
23	4	adsf	2024-02-09	4534	Update Delete

Figure 7: Feedback Management

The screenshot shows a table with two rows of supplier information:

Supplier ID	Supplier Name	Contact Person	Contact Number	Email	Actions
45	rqew	afsdfasfda	5345	afds	Edit Delete
123	tamim	tamim	321	tamim	Edit Delete

Figure 8: Supplier Information

Dashboard	Employee Information	Access Rights	Customer Information	Feedback	Supplier Information	Product Catalog	Price Details	Product Information	Purchase History																					
Sales Transaction	Stock Levels	Task Assignment																												
Product Catalog																														
Add New Product																														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Product ID</th> <th>Product Name</th> <th>Description</th> <th>Category</th> <th>Supplier ID</th> <th colspan="2">Actions</th> </tr> </thead> <tbody> <tr> <td>12</td> <td>newadsfa</td> <td>asdf</td> <td>fasd</td> <td>123</td> <td><button>Edit</button></td> <td><button>Delete</button></td> </tr> <tr> <td>2111</td> <td>pc</td> <td>brand_new</td> <td>first class</td> <td>123</td> <td><button>Edit</button></td> <td><button>Delete</button></td> </tr> </tbody> </table>										Product ID	Product Name	Description	Category	Supplier ID	Actions		12	newadsfa	asdf	fasd	123	<button>Edit</button>	<button>Delete</button>	2111	pc	brand_new	first class	123	<button>Edit</button>	<button>Delete</button>
Product ID	Product Name	Description	Category	Supplier ID	Actions																									
12	newadsfa	asdf	fasd	123	<button>Edit</button>	<button>Delete</button>																								
2111	pc	brand_new	first class	123	<button>Edit</button>	<button>Delete</button>																								

Figure 9: Product Catalog

Dashboard	Employee Information	Access Right	Customer Information	Feedback	Supplier Information	Product Catalog	Price Details	Product Information	Purchase History										
Sales Transaction	Stock Levels	Task Assignment																	
Manage Price Details																			
Add New Price Detail																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Price ID</td> <td>Product ID</td> <td>Price</td> <td>Effective date</td> <td>Actions</td> </tr> <tr> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="button" value="Add Price Detail"/></td> </tr> </table>										Price ID	Product ID	Price	Effective date	Actions	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add Price Detail"/>
Price ID	Product ID	Price	Effective date	Actions															
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add Price Detail"/>															

Figure 10: Manage Price Details

Dashboard	Employee Information	Access Right	Customer Information	Feedback	Supplier Information	Product Catalog	Price Details	Product Information	Purchase History								
Sales Transaction	Stock Levels	Task Assignment															
Manage Product Information																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><input type="button" value="Add Product"/></td> <td>Product Id</td> <td>Product Name</td> <td>Category</td> <td>Price</td> <td>Stock Quantity</td> <td>Supplier Id</td> <td>Actions</td> </tr> </table>										<input type="button" value="Add Product"/>	Product Id	Product Name	Category	Price	Stock Quantity	Supplier Id	Actions
<input type="button" value="Add Product"/>	Product Id	Product Name	Category	Price	Stock Quantity	Supplier Id	Actions										

Figure 11: Manage Product Information

Dashboard	Employee Information	Access Rights	Customer Information	Feedback	Supplier Information	Product Catalog	Price Details	Product Information	Purchase History														
Sales Transaction	Stock Levels	Task Assignment																					
<h2>Purchase History</h2>																							
Add New Purchase																							
<table border="1"> <thead> <tr><th>Transaction ID</th><th>Customer ID</th><th>Product ID</th><th>Purchase ID</th><th>Quantity</th><th>Total Amount</th><th>Actions</th></tr> </thead> <tbody> <tr><td>1</td><td>4534</td><td>12</td><td>1</td><td>2</td><td>123123.00</td><td>Edit Delete</td></tr> </tbody> </table>										Transaction ID	Customer ID	Product ID	Purchase ID	Quantity	Total Amount	Actions	1	4534	12	1	2	123123.00	Edit Delete
Transaction ID	Customer ID	Product ID	Purchase ID	Quantity	Total Amount	Actions																	
1	4534	12	1	2	123123.00	Edit Delete																	

Figure 12: Purchase History

Dashboard	Employee Information	Access Rights	Customer Information	Feedback	Supplier Information	Product Catalog	Price Details	Product Information	Purchase History														
Sales Transaction	Stock Levels	Task Assignment																					
<h2>Sales Transaction</h2>																							
Add New Transaction																							
<table border="1"> <thead> <tr><th>Transaction ID</th><th>Customer ID</th><th>Employee ID</th><th>Transaction Date</th><th>Total Amount</th><th>Payment Method</th><th>Actions</th></tr> </thead> <tbody> <tr><td>1</td><td>4534</td><td>1</td><td>2024-02-08</td><td>123123.00</td><td>online</td><td>Edit Delete</td></tr> </tbody> </table>										Transaction ID	Customer ID	Employee ID	Transaction Date	Total Amount	Payment Method	Actions	1	4534	1	2024-02-08	123123.00	online	Edit Delete
Transaction ID	Customer ID	Employee ID	Transaction Date	Total Amount	Payment Method	Actions																	
1	4534	1	2024-02-08	123123.00	online	Edit Delete																	

Figure 13: Sales Transaction

Dashboard	Employee Information	Access Right	Customer Information	Feedback	Supplier Information	Product Catalog	Price Details	Product Information	Purchase History														
Sales Transaction	Stock Levels	Task Assignment																					
<h2>Manage Stock Levels</h2>																							
Add New Stock Level																							
<table border="1"> <tr><td>Stock ID</td><td>Product ID</td><td>Current Stock</td><td>Minimum Stock</td><td>Maximum Stock</td><td>Last restockdate</td><td>Actions</td></tr> <tr><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td><td>Edit</td></tr> </table>										Stock ID	Product ID	Current Stock	Minimum Stock	Maximum Stock	Last restockdate	Actions	<input type="text"/>	Edit					
Stock ID	Product ID	Current Stock	Minimum Stock	Maximum Stock	Last restockdate	Actions																	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Edit																	

Figure 14: Manage Stock Levels

Task Assignment						
Dashboard		Employee Info		Access Rights		Customer Info
Purchase History		Sales Transaction		Stock Levels		Task Assignment
Add New Task						
Task ID	Employee ID	Description	Deadline	Status	Actions	
12	1	bew	2024-02-09	ht	Edit	Delete
34	1	staff	2024-02-10	processing	Edit	Delete

Figure 15: Task Assignment

8.0 Project Problems and Pitfalls

Learning Curve with XAMPP and SQL : Transitioning to XAMPP and SQL poses a significant learning curve for the team, especially since they are all new to this software stack. SQL, while a standard language can still be complex for beginners to grasp fully and leading to doubts and uncertainties during development.

Lack of Experience with XAMPP : Our team's lack of experience with XAMPP, particularly in building user interfaces can lead to inefficiencies and uncertainties in the development process. Without prior experience, they may struggle with understanding how to effectively convey their application requirements using XAMPP.

Version Compatibility and Updates : XAMPP also undergoes updates and changes over time. This means that the team may encounter compatibility issues or discrepancies between blog tutorials and the current version of XAMPP. Keeping up with these updates and ensuring compatibility with the latest version can be time-consuming and challenging.

Impact on Database Schema : Implementing the user interface using XAMPP after completing the Entity-Relationship Diagram (ERD) may lead to adjustments in the database schema. Certain attributes that were initially included in the ERD may not be necessary for the XAMPP implementation, requiring modifications to align the schema with the interface. This process can be time-consuming and may introduce inconsistencies if not managed properly.

Time-consuming Research and Modifications : The need to conduct extensive research to understand the workings of XAMPP and how it integrates with SQL can consume a significant amount of time. Additionally, modifying the ERD to match the requirements of the XAMPP implementation further adds to the project's time constraints and complexity.

Potential for Inefficient Implementation : Without prior experience or a clear understanding of XAMPP's capabilities, there is a risk of implementing the user interface inefficiently. This could result in suboptimal performance or difficulties in maintaining and updating the application in the future.

Dependency on External Resources : Relying on external resources such as tutorials, forums, and documentation for learning XAMPP and troubleshooting issues may introduce dependencies and delays in the project timeline. Difficulty in finding relevant and up-to-date resources can further exacerbate this challenge.

9.0 Conclusion/Recommendations/Future Works

In conclusion, building a solid database system is a difficult undertaking that requires a great deal of commitment and experience. Our project resulted in the creation of an online AIU Shops system that allows customers and store owners to communicate easily. Users can easily access, edit, and arrange pertinent data with this system. A user-friendly interface is offered to customers so they can explore products, put items in their cart, and arrange for convenient pick-up times. Our database architecture uses complex table structures, inter-table interactions, and enforced restrictions to manage a large amount of data efficiently. Moreover, the solution provides owners with simplified database management capabilities that boost productivity while maintaining data integrity. Notably, improved data consistency is a result of consumers implementing more seamless methods for sharing data.

Our team gained essential information and abilities from this assignment. In order to build the foundation of our system, we used SQL Database, a recently purchased technology. Furthermore, we were able to identify and assess the degree of links between different entities throughout modules by becoming proficient in the production of Entity-Relationship Diagrams, or ERDs. Furthermore, the best possible database organization was guaranteed by our expertise in ERD normalization procedures. We overcame obstacles together and supported one another, which helped to create a positive team environment. This project validated our understanding and skills acquired throughout the course by showing that we could use newly learnt concepts in an efficient manner in a real-world setting.

Future Works

Going forward, we are dedicated to raising the bar for our database system and demonstrating professionalism in order to meet the needs of each and every user. Although our team has worked together to produce a working system that can handle necessary duties, we are aware that there is still much space for development. Our goal is still to improve the system's usefulness through the development of a more user-friendly Graphical User Interface (GUI). In addition, we want to strengthen the security of our database system by introducing an improved login and registration procedure. This program will support data security protocols, giving our users more assurance. Furthermore, our goal is to improve the platform by increasing the range of products and streamlining the checkout process so that customers have a flawless transaction experience.

10.0 References

- 1.Shen, Y. (2010). Design and Implementation of a Web Shop System.
- 2.Islami, R. P., & Mulyanto, A. (2014, November). Component design of business process web content management system for online shop website. In *2014 International Conference on Data and Software Engineering (ICODSE)* (pp. 1-8). IEEE.
- 3.Ruan, Y. (2019). The Development For An Online Shop Platform.