

Comprehensive Notes:

Unit 4: Work Integrated Learning IT – DMA

Ashish Kumar Bhoi
Odisha Adarsha Vidyalaya Bibina

November 13, 2025

Contents

1 Comprehensive Notes	2
1.1 Introduction	2
1.2 Identification of Potential Work Areas	2
1.3 A Shopping Website - A Case Study	2
1.3.1 Entities Involved	2
1.3.2 Entity Relationships	4
1.3.3 Functionality	4
2 Practice Questions	8
2.1 Multiple Choice Questions (MCQs)	8
2.2 One-Word Answer Questions	10
2.3 Differentiate Between Questions	11
2.4 Short Answer Questions	11
2.5 Long Answer Questions	12

1 Comprehensive Notes

1.1 Introduction

This unit explores the practical application of database management concepts. While previous chapters focused on theory and SQL, this unit covers how database management systems (DBMS) are used in real-world work areas. We will analyze a case study of an e-commerce shopping website, "XYZ Shopping Website," to understand its database structure and the application logic (using Java) that manages it.

1.2 Identification of Potential Work Areas

Database Management Systems are essential in any field that needs to store, manage, and retrieve data efficiently. Some of the most common domains include:

- **Education:** Storing student details, marks, and results; managing faculty and staff information; and tracking infrastructure and courses.
- **Banking:** Managing customer information (personal details, accounts, loans), recording daily transactions, and storing employee details.
- **Hospitals:** Maintaining patient health records, managing doctor and staff schedules, and tracking rooms, equipment, and medicines.
- **Government Sector:** Storing electoral rolls, tax records, criminal records, and managing national IDs like PAN cards and AADHAR cards.
- **Companies:** Managing employee information, tracking project details, and monitoring infrastructure, sales, and investments.
- **E-Commerce:** Storing product information (price, quantity), managing customer accounts, and processing orders (as seen in our case study).
- **Airlines & Railways:** Managing flight/train details (schedules, fares), passenger capacity, and tracking online/offline reservations.
- **Telecommunications:** Storing customer details, call records, billing information, and details about subscription plans.
- **Hotels:** Managing guest information, check-in/out times, room availability, and staff/menu details.

1.3 A Shopping Website - A Case Study

We will analyze the "XYZ Shopping Website," a 24x7 online store. It allows customers to browse products, add them to a cart, and pay via options like Cash on Delivery, credit/debit card, or net banking.

1.3.1 Entities Involved

The database for this website is composed of several tables (entities) that store specific pieces of information. The primary key (PK) is the unique identifier for each row in a table. A foreign key (FK) is a field that links to the primary key of another table.

1. **CATEGORY:** Stores the categories of products available.
2. **PRODUCT:** Stores details for every product available.
3. **CUSTOMER:** Stores customer details for login and shipping.
4. **SELLER:** Stores details about the sellers of products.

Table 1: Schema for CATEGORY Table

Field Name	Data Type	Remarks
Category_id	INT(10)	Primary Key (PK)
Name	VARCHAR(20)	Name of the category (e.g., 'Books')
Description	VARCHAR(40)	Description of the category

Table 2: Schema for PRODUCT Table

Field Name	Data Type	Remarks
Product_id	INT(10)	Primary Key (PK)
Name	VARCHAR(20)	Name of the product
Category_id	INT(10)	Foreign Key (FK) links to CATEGORY
Price	INT(10)	Price of the product
Quantity	INT(10)	Number of items in stock
Discount	INT(3)	Discount percentage
Brand	VARCHAR(20)	Brand of the product
Color	VARCHAR(20)	Color of the product
Size	VARCHAR(20)	Size of the product
Seller_id	INT(10)	Foreign Key (FK) links to SELLER
Description	VARCHAR(20)	Description of the product

Table 3: Schema for CUSTOMER Table

Field Name	Data Type	Remarks
Customer_id	INT(10)	Primary Key (PK)
Password	VARCHAR(30)	Login password
First_name	VARCHAR(20)	Customer's first name
Last_name	VARCHAR(20)	Customer's last name
Address	VARCHAR(50)	Customer's shipping address
Email_id	VARCHAR(20)	Customer's email
Contact_num	INT(10)	Customer's contact number

Table 4: Schema for SELLER Table

Field Name	Data Type	Remarks
Seller_id	INT(10)	Primary Key (PK)
Rating	INT(1)	Seller's rating (e.g., 1-5)

Table 5: Schema for WISH_LIST Table

Field Name	Data Type	Remarks
Customer_id	INT(10)	Composite PK, FK links to CUSTOMER
Product_id	INT(10)	Composite PK, FK links to PRODUCT

5. **WISH_LIST**: Stores products a customer wishes to buy later.
6. **ORDER**: Stores details of all placed orders.
7. **SHIPMENT**: Stores shipment details for dispatching products.
8. **PAYMENT**: Stores details of the payment for an order.

Table 6: Schema for ORDER Table

Field Name	Data Type	Remarks
Order_id	INT(10)	Primary Key (PK)
Customer_id	INT(10)	Foreign Key (FK) links to CUSTOMER
Product_id	INT(50)	Foreign Key (FK) links to PRODUCT
Shipment_id	INT(10)	Foreign Key (FK) links to SHIPMENT
Order_date	DATE	Date the order was placed
Current_status	VARCHAR(20)	e.g., 'Pending' or 'Delivered'

Table 7: Schema for SHIPMENT Table

Field Name	Data Type	Remarks
Shipment_id	INT(10)	Primary Key (PK)
Details	VARCHAR(30)	e.g., dispatch location
Shipment_date	DATE	Date the order was shipped

Table 8: Schema for PAYMENT Table

Field Name	Data Type	Remarks
Payment_id	INT(10)	Primary Key (PK)
Order_id	INT(10)	Foreign Key (FK) links to ORDER
Payment_amount	INT(10)	Amount paid
Payment_date	DATE	Date of payment
Mode	VARCHAR(20)	e.g., 'Credit Card', 'Cash on Delivery'

1.3.2 Entity Relationships

The tables are all linked using their Primary and Foreign Keys. This diagram (based on the schemas) shows how the entities are related.

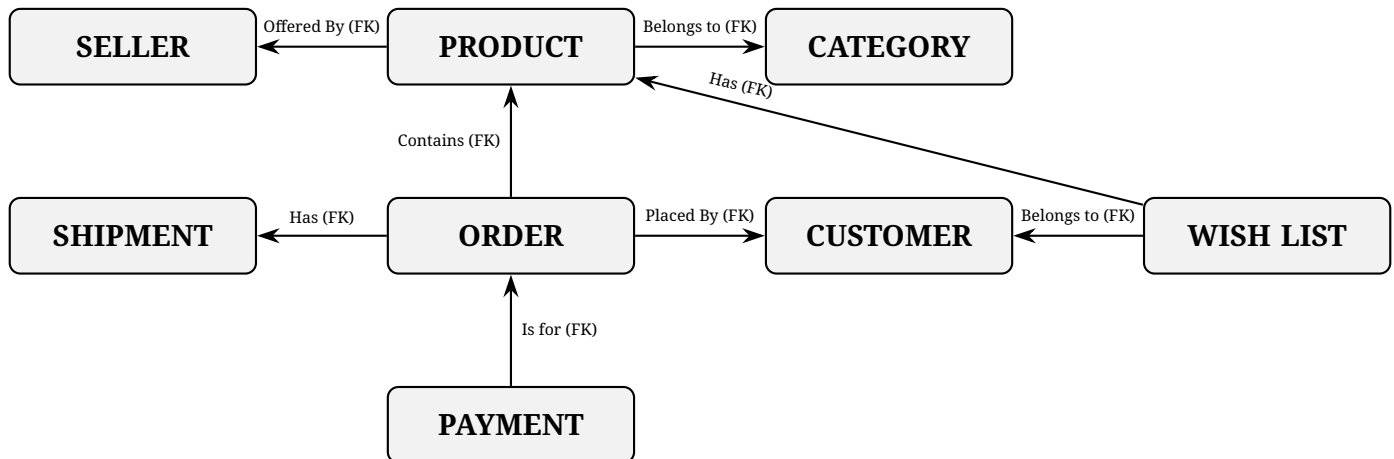


Figure 1: Relationships Between Entities (Based on Schema)

1.3.3 Functionality

The website uses Java classes with specific functions (methods) to interact with the database. These functions use SQL commands (like 'INSERT', 'DELETE', 'UPDATE', 'SELECT') to manage the data.

1. **Managing Categories:** The Category class manages the CATEGORY table.

```

1 package shopping_application;
2 public class Category {
3     public void insert_category(String categ_tuple[]) {

```

```

4      // Add functionality to insert a row of category
5      // named categ_tuple.
6      // SQL command to be used: insert
7  }
8  public void delete_category(int categ_id) {
9      // Add functionality to delete a category with id categ_id.
10     // SQL command to be used: delete
11 }
12 public void modify_category(int categ_id, String attr, String new_val) {
13     // Add functionality to change value of attribute attr
14     // of category with id categ_id to new_val.
15     // SQL command to be used: update
16 }
17 public void display_category(int categ_id) {
18     // Add functionality to display all the products in
19     // category with id categ_id.
20     // SQL command to be used: select
21 }
22 }

```

Listing 1: Class for Table CATEGORY

2. Managing Products: The Product class manages the PRODUCT table.

```

1 package shopping_application;
2 public class Product {
3     public void insert_product(String prod_tuple[]) {
4         // Add functionality to insert a row of product named
5         // prod_tuple.
6         // SQL command to be used: insert
7     }
8     public void delete_product(int prod_id) {
9         // Add functionality to delete a product with id prod_id.
10        // SQL command to be used: delete
11    }
12    public void modify_product(int prod_id, String attr, String new_val) {
13        // Add functionality to change value of attribute attr
14        // of product with id prod_id to new_val.
15        // SQL command to be used: update
16    }
17    public void display_product(int prod_id) {
18        // Add functionality to display details of the products
19        // with id prod_id
20        // SQL command to be used: select
21    }
22 }

```

Listing 2: Class for Table PRODUCT

3. Managing Sellers: The Seller class manages the SELLER table.

```

1 package shopping_application;
2 public class Seller {
3     public void insert_seller(String seller_tuple[]) {
4         // Add functionality to insert a row of seller named
5         // seller_tuple.
6         // SQL command to be used: insert
7     }
8     public void delete_seller(int seller_id) {
9         // Add functionality to delete a seller with id seller_id.
10        // SQL command to be used: delete
11    }
12    public void modify_seller(int seller_id, String attr, String new_val) {
13        // Add functionality to change value of attribute attr

```

```

14      // of seller with id seller_id to new_val.
15      // SQL command to be used: update
16  }
17  public void display_seller(int seller_id) {
18      // Add functionality to display details of the seller
19      // with id seller_id
20      // SQL command to be used: select
21  }
22 }

```

Listing 3: Class for Table SELLER

4. Managing Customers: The Customer class manages the CUSTOMER table.

```

1 package shopping_application;
2 public class Customer {
3     public void insert_customer(String customer_tuple[]) {
4         // Add functionality to insert a row of customer named
5         // customer_tuple.
6         // SQL command to be used: insert
7     }
8     public void modify_customer(int customer_id, String attr, String new_val) {
9         // Add functionality to change value of attribute attr
10        // of customer with id customer_id to new_val.
11        // SQL command to be used: update
12    }
13 }

```

Listing 4: Class for Table CUSTOMER

5. Managing Wish List: The WishList class manages the WISH_LIST table.

```

1 package shopping_application;
2 public class WishList {
3     public void insert_wish(String wish_tuple[]) {
4         // Add functionality to insert a row of wish named
5         // wish_tuple.
6         // SQL command to be used: insert
7     }
8     public void delete_wish(int customer_id, String wish) {
9         // Add functionality to delete a wish of customer with id
10        // customer_id.
11        // SQL command to be used: delete
12    }
13     public void display_wish(int customer_id) {
14         // Add functionality to display wish list of customer with
15         // id customer_id.
16         // SQL command to be used: select
17     }
18 }

```

Listing 5: Class for Table WISH_LIST

6. Managing Payments: The Payment class manages the PAYMENT table. This is used when an order is placed.

```

1 package shopping_application;
2 public class Payment {
3     public void insert_payment(String payment_tuple[]) {
4         // Add functionality to insert a row of payment named
5         // payment_tuple.
6         // SQL command to be used: insert
7     }
8 }

```

Listing 6: Class for Table PAYMENT

7. Managing Orders: The Order class manages the ORDER table.

```
1 package shopping_application;
2 public class Order {
3     public void insert_order(String order_tuple[]) {
4         // Add functionality to insert a row of order named
5         // order_tuple.
6         // SQL command to be used: insert
7     }
8     public void delete_order(int order_id) {
9         // Add functionality to delete order with id order_id.
10        // SQL command to be used: delete
11    }
12    public void display_order(int order_id) {
13        // Add functionality to display details of the order with
14        // id order_id.
15        // SQL command to be used: select
16    }
17 }
```

Listing 7: Class for Table ORDER

8. Managing Shipments: The Shipment class manages the SHIPMENT table.

```
1 package shopping_application;
2 public class Shipment {
3     public void insert_shipment_details(String order_shipment[]) {
4         // Add functionality to insert a row of shipment named
5         // order_shipment.
6         // SQL command to be used: insert
7     }
8 }
```

Listing 8: Class for Table SHIPMENT

2 Practice Questions

2.1 Multiple Choice Questions (MCQs)

1. Which of the following is NOT a domain where DBMS is used?

- | | |
|--------------|----------------------------------|
| (A) Banking | (C) Hospitals |
| (B) Airlines | (D) A simple notepad application |

Answer: (D)

2. In the XYZ Shopping Website, which table stores product price and quantity?

- | | |
|--------------|--------------|
| (A) CATEGORY | (C) CUSTOMER |
| (B) PRODUCT | (D) ORDER |

Answer: (B)

3. What is the primary key for the WISH_LIST table?

- | | |
|--|------------------------------------|
| (A) Customer_id only | Product_id |
| (B) Product_id only | |
| (C) A composite key of Customer_id and | (D) It does not have a primary key |

Answer: (C)

4. In the PRODUCT table, Category_id is an example of a:

- | | |
|-----------------|-------------------|
| (A) Primary Key | (C) Composite Key |
| (B) Foreign Key | (D) Super Key |

Answer: (B)

5. Which table would store a customer's login password?

- | | |
|------------|--------------|
| (A) ORDER | (C) PAYMENT |
| (B) SELLER | (D) CUSTOMER |

Answer: (D)

6. A customer wants to see all products in the "Electronics" category. Which function would be invoked?

- | | |
|----------------------|---------------------|
| (A) display_category | (C) insert_category |
| (B) display_product | (D) modify_product |

Answer: (A)

7. Which SQL command would the insert_customer function most likely use?

- | | |
|------------|------------|
| (A) SELECT | (C) INSERT |
| (B) UPDATE | (D) DELETE |

Answer: (C)

8. To store the details of an order, like the date it was placed, which table is used?

- (A) PAYMENT
- (B) SHIPMENT

- (C) ORDER
- (D) CUSTOMER

Answer: (C)

9. A function named `modify_seller` would use which SQL command to change a seller's rating?

- (A) UPDATE
- (B) MODIFY

- (C) SELECT
- (D) INSERT

Answer: (A)

10. Which table links an order to the payment made for it?

- (A) The ORDER table contains a `Payment_id`.
- (B) The PAYMENT table contains an `Order_id`.

- (C) The CUSTOMER table links them.
- (D) A separate junction table is used.

Answer: (B)

11. What is the purpose of the `WISH_LIST` table?

- (A) To store products the customer has bought.
- (B) To store products the customer has in their cart.
- (C) To store products the customer wants to

- buy later.
- (D) To store all products the seller wishes to sell.

Answer: (C)

12. Which function would be called when a user wants to remove an order they just placed?

- (A) `delete_order`
- (B) `delete_customer`

- (C) `modify_order`
- (D) `display_order`

Answer: (A)

13. In a hospital DBMS, which information would likely be stored?

- (A) Patient health records.
- (B) Flight arrival times.

- (C) Customer bank account numbers.
- (D) Product quantity in stock.

Answer: (A)

14. The `Seller_id` in the `PRODUCT` table implies what relationship?

- (A) One seller can offer many products.
- (B) One product can be offered by many sellers.

- (C) One seller can only offer one product.
- (D) This is not a relationship.

Answer: (A)

15. Which table would store the dispatch date and location for an order?

- (A) ORDER
- (B) PAYMENT

- (C) SHIPMENT
- (D) SELLER

Answer: (C)

16. The Java class `Customer` is used to manage which table?

- (A) ORDER
- (B) CUSTOMER

- (C) SELLER
- (D) CATEGORY

Answer: (B)

17. What SQL command is used by the `display_order` function?

- (A) INSERT
- (B) UPDATE

- (C) DELETE
- (D) SELECT

Answer: (D)

18. Which table stores the 'Mode' of payment (e.g., 'Cash on Delivery')?

- (A) ORDER
- (B) CUSTOMER

- (C) PAYMENT
- (D) SHIPMENT

Answer: (C)

2.2 One-Word Answer Questions

1. A domain for storing student details, marks, and results. (*Education*)
2. A domain for storing flight details and reservations. (*Airlines*)
3. A domain for storing customer accounts and daily transactions. (*Banking*)
4. The table that stores product types (like 'Books') in the case study. (*CATEGORY*)
5. The table that stores customer names, addresses, and emails. (*CUSTOMER*)
6. The table that stores information about orders that have been placed. (*ORDER*)
7. The table that stores details of the seller, like their rating. (*SELLER*)
8. The key that uniquely identifies a row in the `PRODUCT` table. (*Product_id / Primary Key*)
9. The `Customer_id` field in the `ORDER` table is an example of this type of key. (*Foreign Key*)
10. The SQL command used to add a new row to a table. (*INSERT*)
11. The SQL command used to change existing data in a table. (*UPDATE*)
12. The SQL command used to remove a row from a table. (*DELETE*)
13. The SQL command used to retrieve or view data. (*SELECT*)
14. A primary key made of two or more columns, like in `WISH_LIST`. (*Composite Key*)
15. The table that stores information about "Cash on Delivery" or "Credit Card". (*PAYMENT*)
16. The table that stores details like "dispatched from warehouse". (*SHIPMENT*)
17. The function to add an item to a customer's "buy later" list. (*insert_wish*)
18. The function to track the status of an order. (*display_order*)
19. The Java class used to manage the `PRODUCT` table. (*Product*)
20. The field in the `ORDER` table that tracks if an order is 'Pending' or 'Delivered'. (*Current_status*)

2.3 Differentiate Between Questions

1. CATEGORY Table and PRODUCT Table

Answer: The CATEGORY table stores the general types or classifications of products (e.g., 'Books', 'Electronics'). The PRODUCT table stores the specific, individual items within those categories (e.g., 'Laptop Model X', 'Science Textbook'). The PRODUCT table uses a foreign key (Category_id) to link to the CATEGORY table.

2. CUSTOMER Table and SELLER Table

Answer: The CUSTOMER table stores information about the end-users who buy products from the website (e.g., their name, address, password). The SELLER table stores information about the people or companies that list and sell products *through* the website (e.g., their rating).

3. ORDER Table and PAYMENT Table

Answer: The ORDER table stores high-level information about the order itself (e.g., who placed it, when it was placed, what its status is, and which products/shipment are involved). The PAYMENT table stores information specifically about the financial transaction for that order (e.g., how much was paid, the date of payment, and the mode of payment like 'Credit Card'). PAYMENT links to ORDER via a foreign key.

4. WISH_LIST Table and ORDER Table

Answer: The WISH_LIST table stores products that a customer is interested in but has *not* yet purchased. It's a "save for later" list. The ORDER table stores a record of products that a customer has actually committed to buying and has placed an order for.

5. Primary Key and Foreign Key

Answer: A Primary Key is a field that uniquely identifies each record in its *own* table (e.g., Product_id in the PRODUCT table). A Foreign Key is a field in one table that refers to the Primary Key of *another* table, creating a link or relationship (e.g., Seller_id in the PRODUCT table links to the SELLER table's primary key).

6. insert_product and modify_product functions

Answer: The insert_product function uses the SQL INSERT command to add a completely new product row to the PRODUCT table. The modify_product function uses the SQL UPDATE command to change data in an *existing* product's row (e.g., to change its price or quantity).

2.4 Short Answer Questions

1. List three different work areas (besides e-commerce) where DBMS is used.

Answer: Three domains are: **Education** (for storing student marks and details), **Banking** (for storing customer transactions and account information), and **Hospitals** (for maintaining patient health records).

2. In the case study, what is the purpose of the CUSTOMER table?

Answer: The CUSTOMER table's purpose is to store all personal and account details for a customer, such as their name, address, contact number, and password. This information is essential for user login and for shipping orders to the correct address.

3. Why is a separate SHIPMENT table necessary?

Answer: A separate SHIPMENT table is good database design. It stores specific details about the dispatch process, like the shipment date and location. An order might also be split into multiple shipments, which a separate table can handle easily (the current schema doesn't, but this design allows for it).

4. What is the relationship between the ORDER, CUSTOMER, and PRODUCT tables?

Answer: The ORDER table links the other two. It contains a Customer_id (as a foreign key) to identify *who* placed the order, and a Product_id (as a foreign key) to identify *what* was ordered.

5. What happens in the database when a user calls the insert_wish function?

Answer: When insert_wish is called, the application executes an INSERT SQL command. This

command adds a new row to the WISH_LIST table, containing the customer's Customer_id and the Product_id of the item they wish to save.

6. **List two functions from the Category class and their purpose.**

Answer: Two functions are insert_category and delete_category. insert_category is used to add a new product category (like 'Furniture') to the CATEGORY table. delete_category is used to remove a category from the table.

7. **Why does the WISH_LIST table have a composite primary key?**

Answer: It has a composite key of (Customer_id, Product_id) to uniquely identify a row. This is because a customer can have many products in their wish list, and a product can be in many customers' wish lists. The combination of both IDs is needed to represent one unique "wish".

8. **How does the database handle a "Cash on Delivery" payment?**

Answer: When a customer chooses "Cash on Delivery," a record is still created in the PAYMENT table with the mode set to 'Cash on Delivery'. However, the Payment_date attribute is likely set to the date the product is delivered, not the date the order was placed.

9. **What is the purpose of the display_order function in the Order class?**

Answer: The display_order function is used to track an order. It would run a SELECT SQL command to retrieve all details for a specific order_id, including its Current_status, and display them to the user.

10. **What is the purpose of the SELLER table?**

Answer: The SELLER table stores information about the different vendors who sell products on the website. This allows the website to be a marketplace (like Amazon) and track seller-specific information, such as their Rating.

2.5 Long Answer Questions

1. **Explain the full process of a new customer registering and then placing their first order. Mention all key tables and Java functions involved.**

Answer: The process involves two main phases:

(a) **Registration:** The new customer fills out the "Sign Up" form. This action calls the insert_customer function from the Customer class. This function executes an INSERT SQL command, adding a new row to the CUSTOMER table with the customer's First_name, Address, Password, etc.

(b) **Placing an Order:**

- The customer (now logged in) adds a product to their cart and proceeds to checkout.
- They choose a payment option (e.g., 'Credit Card'). This triggers the insert_payment function from the Payment class, which adds a row to the PAYMENT table with the amount, mode, and date.
- The system then calls the insert_shipment_details function from the Shipment class, creating a row in the SHIPMENT table to prepare for dispatch.
- Finally, the insert_order function from the Order class is called. This creates a row in the ORDER table, which links everything together. This new row stores the Order_id, the Customer_id (from the logged-in user), the Product_id, and the new Shipment_id. It also sets the Current_status to 'Pending'.

2. **Describe the database schema for the PRODUCT, CATEGORY, and SELLER tables. Explain the relationships between them.**

Answer:

- **CATEGORY Table:** This is a simple table with fields like Category_id (Primary Key) and Name. It acts as a lookup table for product types.
- **SELLER Table:** This is another simple table with fields like Seller_id (Primary Key) and Rating. It acts as a lookup table for vendors.

- **PRODUCT Table:** This is the central table. Its Primary Key is `Product_id`. It contains descriptive fields like Name, Price, and Quantity.
- **Relationships:** The PRODUCT table is linked to the other two tables via foreign keys.
 - It has a `Category_id` field, which is a Foreign Key that points to the `Category_id` in the CATEGORY table. This creates a **one-to-many** relationship (one category can have many products).
 - It also has a `Seller_id` field, which is a Foreign Key that points to the `Seller_id` in the SELLER table. This creates a **one-to-many** relationship (one seller can offer many products).

3. **Select two work areas (e.g., Banking, Airlines) and explain in detail what kind of data would be stored in their databases and why it is useful.**

Answer:

- **1. Banking:**
 - **Data Stored:** A banking DBMS would store tables for **Customers** (personal details, address, PAN card), **Accounts** (account number, type, balance, FK to Customer), **Transactions** (transaction ID, date, amount, type, FK to Account), **Loans** (loan ID, amount, interest rate, FK to Customer), and **Employees** (employee ID, name, salary, branch).
 - **Usefulness:** This data is crucial for all banking operations. It allows the bank to manage customer balances, process deposits and withdrawals, calculate interest, track loan repayments, and detect fraudulent activity. It provides a centralized, secure, and fast way to manage all financial data.
- **2. Airlines:**
 - **Data Stored:** An airline DBMS would store tables for **Passengers** (passenger ID, name, passport details), **Flights** (flight number, aircraft type, origin, destination), **Schedules** (schedule ID, flight number, departure time, arrival time), **Reservations** (reservation ID, FK to Passenger, FK to Schedule, seat number), and **Aircraft** (aircraft ID, model, capacity).
 - **Usefulness:** This database is essential for managing all airline logistics. It handles on-line and offline bookings, ensures flights are not overbooked, manages passenger manifests, tracks aircraft maintenance schedules, and sets ticket prices.

4. **What is the difference between the WISH_LIST table and the ORDER table? Explain the purpose and schema of each.**

Answer:

- **WISH_LIST Table:**
 - **Purpose:** Its purpose is to allow customers to "save" products they are interested in but not ready to buy. It's a "buy later" feature.
 - **Schema:** It has a composite primary key of (`Customer_id`, `Product_id`). This links a specific customer to a specific product. It contains no other information.
- **ORDER Table:**
 - **Purpose:** Its purpose is to store a permanent record of a completed transaction. This is the official record that a customer has agreed to purchase a product.
 - **Schema:** It has its own primary key (`Order_id`). It contains multiple foreign keys (`Customer_id`, `Product_id`, `Shipment_id`) to link all the details of the transaction. It also contains important status information like `Order_date` and `Current_status`.
- **Difference:** The key difference is intent. WISH_LIST represents a *potential* purchase with no obligation. ORDER represents an *actual*, confirmed purchase that needs to be processed, paid for, and shipped.

5. **Explain the eight Java classes (e.g., Category, Product, etc.) mentioned in the case study. What is their general purpose and what three SQL commands do they most commonly use?**

Answer: The eight Java classes (Category, Product, Customer, Seller, WishList, Order, Shipment, Payment) are part of the application logic.

- **General Purpose:** Each class acts as a "manager" for its corresponding database table. For example, the `Product` class contains all the functions needed to interact with the `PRODUCT` table. This organizes the code, so instead of writing raw SQL everywhere, the application can just call a function like `Product.delete_product(123)`.
- **Common SQL Commands:**
 - (a) **INSERT:** Used by functions like `insert_category`, `insert_product`, and `insert_order` to add new rows of data to the tables.
 - (b) **UPDATE:** Used by functions like `modify_category` and `modify_customer` to change data in existing rows.
 - (c) **DELETE:** Used by functions like `delete_category` and `delete_order` to remove rows from the tables.
- (A fourth common command is **SELECT**, used by functions like `display_category` and `display_order` to retrieve and show data to the user.)