Amazon Database System Design (Retail)

Course Number: CS 6360.002

Team Number: 06

Team Members:

Kushagra Dar (kxd180025)

Ruchi Singh (rxs180057)

Anant P Srivastava (aps180006)

Introduction:

The goal of the project is to design and implement a retail based relational database that showcases how various entities are related to retail and commerce. It also tries to cover all the relative scenarios that one can come across in the field of e-commerce such as creating invoice, placing order, method for payment etc. In order to implement, Oracle DB is used as the relational database management system (RDBMS).

Requirements gathering & Design Approach:

In order to understand how an e-commerce giant like amazon works, a detailed research was required. The first phase involved identifying the entities in an e-commerce business and how do they interact with each other. So, after a brief understanding, below were some main processes that made up e-commerce:

- A seller selling a product
- A buyer placing order
- The purchased item being supplied by Supplier
- A buyer can have Shopping cart and Wishlist
- A buyer can provide reviews for a product
- A product can have offers
- A purchased product is a part of an order
- An order is placed after successful payment
- An invoice is generated for an order
- A product's quantity is updated after each successfully placed order

Based on above processes and flows, we identified the system is made up of 4 major components:

1. ***User:*** Here, the user is the target of e-commerce for whom the system is built. The user can be a buyer as well as seller. The database stores the user's data and addresses. It also maintains user's Shopping cart and Wishlist. It also maintains the reviews of a product by any user.
2. ***Product:*** For a product, the database stores its name, ID, quantity, what offers are applicable on it, which order it is part of etc. It also stores the reviews of a particular product. A buyer can purchase multiple products and a product can be purchased by many buyers.
3. ***Order:*** An order is a link between a buyer, product and shipper. A buyer places an order for a particular product and shipper ships the particular order to the buyer. On creation of successful order, an invoice is also stored in the database.
4. ***Payment:*** Payment is associated to an order. It stores the payment method for an order and also stores the payment method such as gift card or payment card (credit/debit).

Considering the major components of the above system, we created major entities that could capture data for users, products, orders and payments.

- **User**: Details of user such as User ID, User type, Date created, address of User.
    - **Buyer**: Details of buyer such as buyer ID, name, membership, phone, email.
    - **Seller**: Details of seller such as seller ID, name, seller's company name, phone, email, logo.
    - **Address**: Details of addresses of a particular user such as address ID, address line, city, country, postal code.
- **Shopping Cart and Wish List:** A buyer has a shopping cart (shopping cart ID, quantity) and Wish List (Wish List ID) which contains products.
- **Reviews:** A buyer gives reviews which includes details such as ratings, customer review, review ID for a product.
- **Product:** Details of products such as product ID, product name, department, quantity (units in stock, units in order), unit price, product description, item picture.
    - **Offers:** Details of offers related to a product such as offer ID, offer amount.
- **Order:** Details of orders such as Order ID, order date, price, item quantity, transaction status, tax, required date, payment date.
    - **Invoice:** Details of invoice generated such as invoice ID, invoice type, invoice amount related to an order.
    - **Shipper:** Details of shipper such as shipper ID, shipper company name, contact name, phone who ships the order to the buyer.
- **Payment:** Details of payment method used for an order. It stores details such as payment date, payment type.

- **Payment Gift Card:** Details of payment gift card used in the payment of an order such as gift card ID, gift card number, gift card amount, gift card expiration month and gift card expiration year.
- **Payment Card:** Details of payment card (debit/credit) used in the payment of an order such as card ID, card number, card expiration month, card expiration year.

In addition to above, enhancements and transaction events such as check product's quantity before adding to cart, decrease the product's quantity after placing the order etc., need to be created to make the system functional and preserve the integrity of the database system. In order to achieve that, triggers and stored procedures are used.

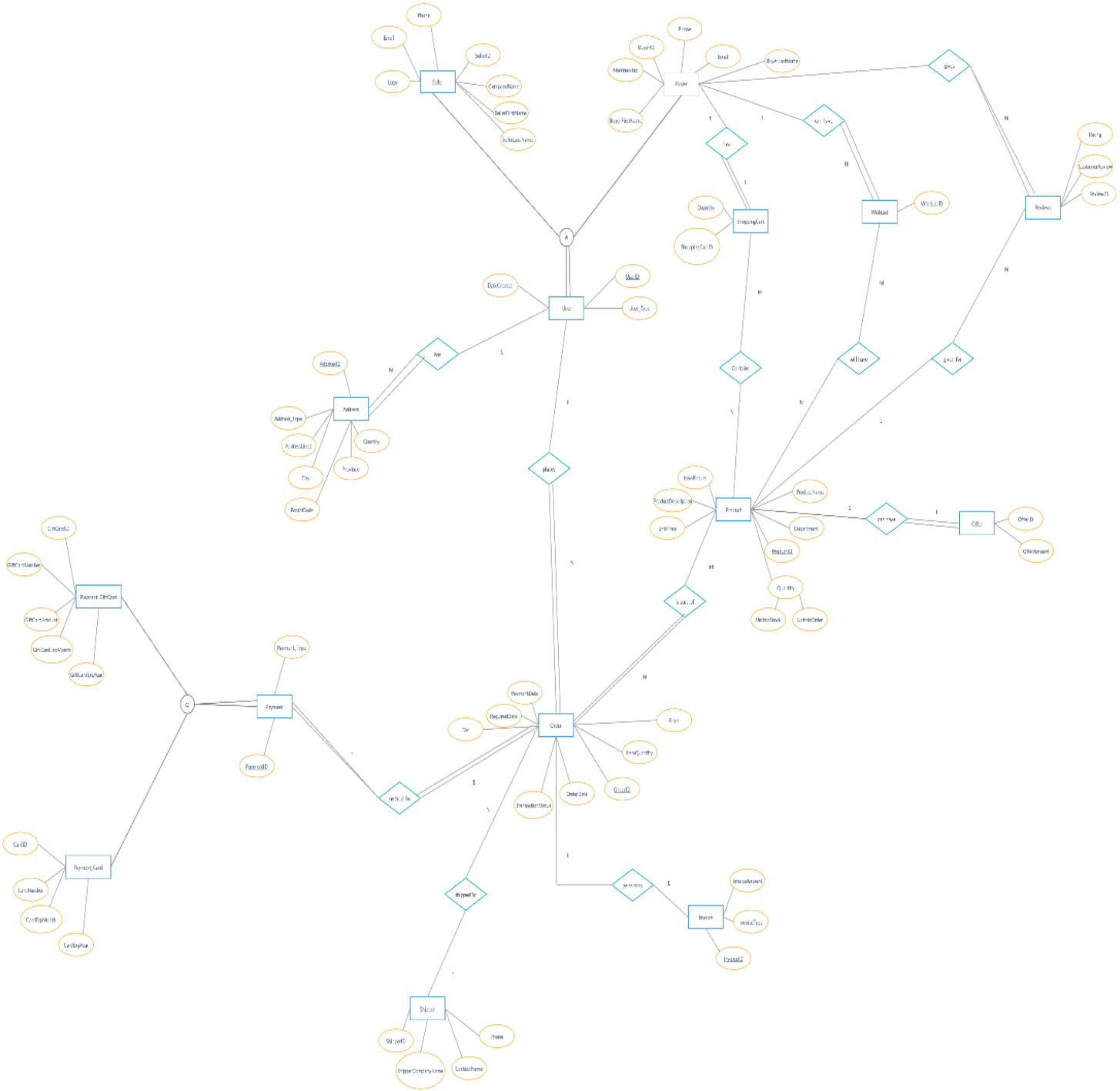
EER Diagram:

To understand the design better, ER diagram provides the conceptual schematic representation of the database system. But before making the ER diagram, one also needs to understand the relationships and cardinality between various entities:

- **User-Address (1-N):** A user may have multiple address and one address must belong to one user.
- **Buyer-Shopping Cart (1-1):** A buyer may have one shopping cart and one shopping cart must belong to one buyer.
- **Buyer-Wish List (1-N):** A buyer may have multiple Wish lists and one Wish list must belong to one buyer.
- **Buyer-Reviews (1-N):** A buyer may give multiple reviews for different products and a review must belong to one buyer.
- **Shopping Cart-Product (M-N):** A shopping cart may contain multiple products and a product may be in different shopping carts.
- **Wish List-Product (M-N):** A wish list may contain multiple products and a product may be in different wish lists.
- **Reviews-Product (N-1):** A review may belong to one product and one product may have multiple reviews.
- **Product-Offers (1-1):** A product may have one offer and an offer must belong to one product.
- **User-Order (1-N):** A user may place multiple orders and an order must belong to one user.
- **Product-Order (M-N):** A product may be a part of multiple orders and an order must contain one or multiple products.
- **Order-Payment (1-1):** An order should have a payment method and a payment method must belong to an order.
- **Order-Shipper (N-1):** An order may have one shipper and a shipper may have multiple orders to ship.

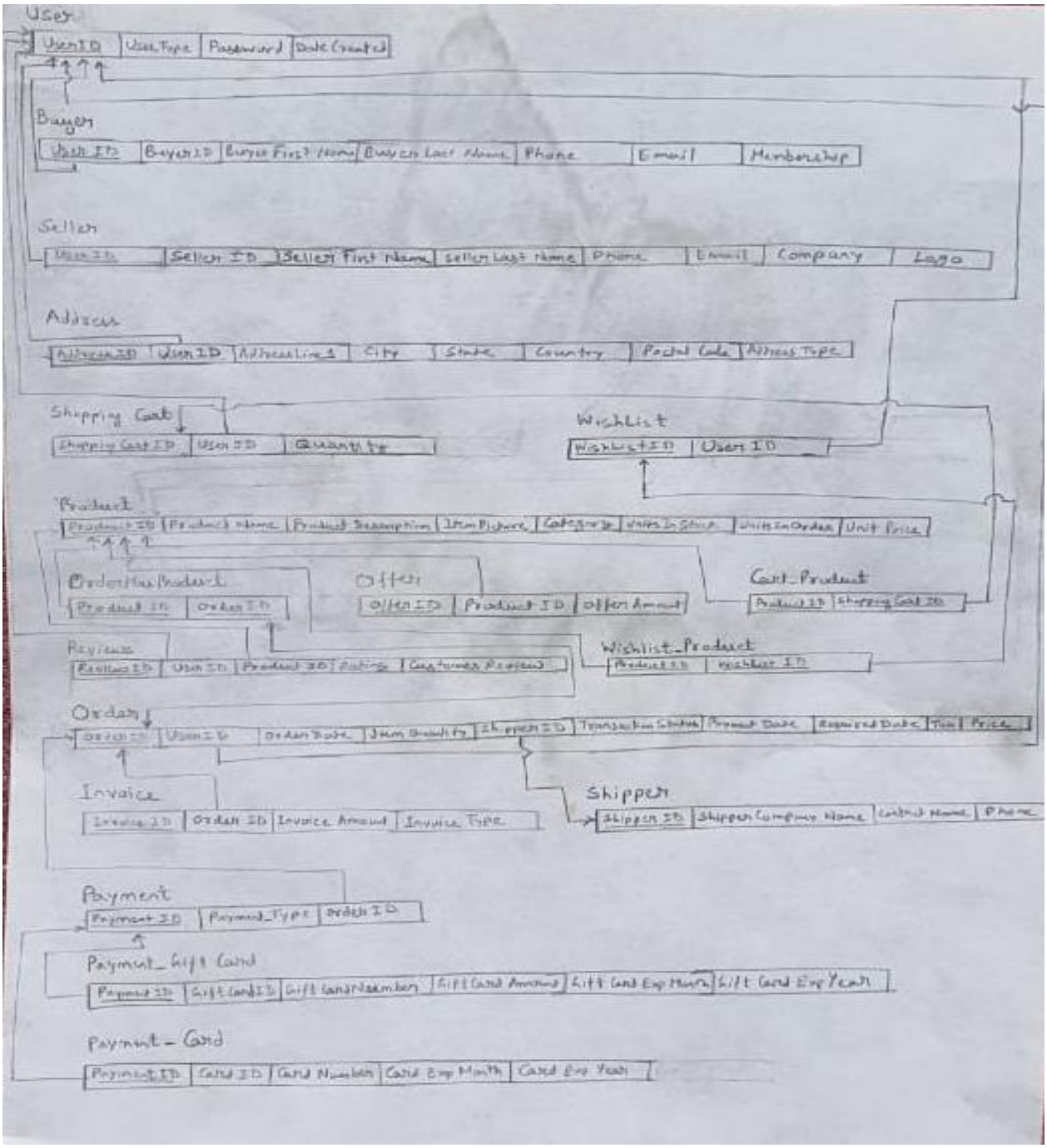
- **Order-Invoice (1-1):** An order may have one invoice and an invoice may belong to one order.

Also, a user has a ***disjoint*** relationship with buyer and seller as a user must only be a buyer or a seller. In addition to above, a payment has an ***overlapping*** relationship with payment gift card and payment card as any of the two or both must be used for payment.



Mapping to Relational Model:

In order to represent database as a collection of relations, relational model representation is used. All the entities are mapped to a relation (table). Keys and relationships are also mapped accordingly. After all the steps were followed in converting EER diagram to relational model, the relational model generated is normalized in 3NF because there are no functional dependencies except between the primary keys and its attributes (under same relation).



SQL (Structured Query Language):

After analyzing the EER diagram and relational model, SQL was used to implement the dynamics and structure of e-commerce database. Following is the link:

https://drive.google.com/file/d/1NEwdVgQfEV0EBr0Dr4a-Yp_8XJhXRVPe/view?usp=sharing

The file '*Amazon.sql*' contains schema of amazon database and its related tables.

PL/SQL (Procedural Language extensions to SQL):

Procedure and triggers are created to insert data into tables.

Stored procedure is a set of SQL statements with an assigned name, which are stored in RDBMS as a group, so that it can be reused and shared by multiple programs. Some procedures created are:

- *Register buyer*: to register the buyer
- *Register seller*: to register the seller
- *Store Product*: to add products in inventory
- *Add Address*: to add the addresses of a user
- *Insert into Cart*: to insert the products into cart
- *Insert Order Product*: to assign which products are assigned to an order ID
- *Place Order*: to place an order
- *Create Invoice*: to generate invoice for placed orders
- *Insert Payment Card*: to insert payment method for card(debit/credit)
- *Insert Payment Gift Card*: to insert payment method for gift card

Trigger is a database object that is associated to a table. It is activated when a defined action is executed for the table. Some triggers created are:

- *Check offer amount:* to restrict the amount of offer than a set amount while inserting
- *Check cart quantity:* to restrict the quantity of products in shopping cart
- *Cart drop:* to remove products from cart as the product is moved to orders
- *Change order status:* to change the order status from 'Pending' to 'Success'
- *Check inventory:* to check whether the product added to cart is out of stock or not

One can find all the stored procedure and triggers in the following link under the file names '*Amazon_Procedures.sql*' and '*Amazon_triggers.sql*':

https://drive.google.com/file/d/1NEwdVgQfEV0EBr0Dr4a-Yp_8XJhXRVPe/view?usp=sharing

Results:

1. Adding a buyer and seller to the database that updates Buyer table and Seller table:

```
BEGIN
   REGISTERBUYER(1, 'b', SYSDATE, 12345678, 'Y', 'JANE', 'DOE', '123-456-7890', 'jdoe@gmail.com');
END;
BEGIN
   REGISTERSELLER(456, 'S', SYSDATE-1, 2, 'AMAZON', 'ANANT', 'PRAKASH', '469-350-1307', 'ruchi@gmail.com', NULL);
END;
```

Buyer table:

	BUYERID	USERID	MEMBERSHIP	BUYERFIRSTNAME	BUYERLASTNAME	PHONENUMBER	EMAIL
1	12345678	1	Y	JANE	DOE	123-456-7890	jdoe@gmail.com

Seller table:

	SELLERID	USERID	COMPANYNAME	SELLERFIRSTNAME	SELLERLASTNAME	PHONENUMBER	EMAIL	LOGO
1	2	456	AMAZON	ANANT	PRAKASH	469-350-1307	ruchi@gmail.com	(null)

It also updates the User table with user ID and user type:

	USERID	USER_TYPE	DATECREATED
1	1	b	03-DEC-19
2	456	S	02-DEC-19

2. Adding products to product table:

```
BEGIN
   StoreProduct(1, 'Electronics', 'Tablet', 5000, 'tablet', 100, 0, NULL);
   StoreProduct(2, 'Sports', 'Football', 100, 'Signed by Messi', 100, 0, NULL);
   StoreProduct(3, 'Kitchen', 'SilverWare', 50, 'kitchen ware', 100, 0, NULL);
   StoreProduct(4, 'Education', 'DB Design Book', 100, 'Study Well', 100, 0, NULL);
END;
```

Product table:

	PRODUCTID	DEPARTMENT	PRODUCTNAME	UNITPRICE	PRODUCTDESCRIPTION	UNITSINSTOCK	UNITSINORDER	ITEMPICTURE
1	1	Electronics	Tablet	5000	tablet	100	0	(null)
2	2	Sports	Football	100	Signed by Messi	100	0	(null)
3	3	Kitchen	SilverWare	50	kitchen ware	100	0	(null)
4	4	Education	DB Design Book	100	Study Well	100	0	(null)

3. Adding address of user to address table:

```
BEGIN
   ADD_ADDRESS('H1', 1, 'Home', '7815 McCallum Blvd.', 'Dallas', 'Texas', 'US', 75252);
   ADD_ADDRESS('W1', 1, 'Work', '2.104, ECSS, UT Dallas', 'Dallas', 'Texas', 'US', 75252);
END;
```

Address table:

	ADDRESSID	USERID	ADDRESS_TYPE	ADDRESSLINE1	CITY	PROVINCE	COUNTRY	POSTALCODE
1	H1	1	Home	7815 McCallum Blvd.	Dallas	Texas	US	75252
2	W1	1	Work	2.104, ECSS, UT Dallas	Dallas	Texas	US	75252

4. Adding products to shopping cart will update shopping cart and cart_product table:

```
BEGIN
    Insert_into_Cart('SH1', 1, 1);
    Insert_into_Cart('SH1', 2, 1);
END;
```

Cart_product table holds the shopping cart ID and Product ID:

	SHOPPINGCARTID	PRODUCTID
1	SH1	1
2	SH1	2

Shopping cart table:

	SHOPPINGCARTID	USERID	QUANTITY
1	SH1	1	2

5. Placing an order will create a record in Orders table and Order_Product table:

```
BEGIN
    Place_Order(999,1,'BlueDart');
END;
```

Order_Product table holds the order ID and product ID:

	ORDERID	PRODUCTID
1	999	1
2	999	2

Orders table (initially transaction status is ‘Pending’ as the payment is not completed for the particular order):

	ORDERID	USERID	SHIPPERID	ORDERDATE	REQUIREDDATE	TAX	TRANSACTIONSTATUS	PAYMENTDATE	ITEMQUANTITY	PRICE
1	999	1	BlueDart	03-DEC-19	10-DEC-19	510	PENDING	03-DEC-19	2	5100

Also, the product is deleted from shopping_cart table and cart_product table.

6. Payment of an order:

```
BEGIN
   Insert_Payment_CARD('XYZ', 999, 'C', '343252', '34456678', '12', '2023');
END;
```

It updates the payment table and payment_card table(if the payment is done via card):

	PAYMENTID	ORDERID	PAYMENT_TYPE
1	XYZ	999	C

	CARDID	PAYMENTID	CARDNUMBER	CARDEXPMONTH	CARDEXPYEAR
1	343252	XYZ	34456678	12	2023

After successful completion of the payment, the transaction status in orders table is changed from 'Pending' to 'Success'.

	ORDERID	USERID	SHIPPERID	ORDERDATE	REQUIREDDATE	TAX	TRANSACTIONSTATUS	PAYMENTDATE	ITEMQUANTITY	PRICE
1	999	1	BlueDart	03-DEC-19	10-DEC-19	510	SUCCESS	03-DEC-19	2	5100

7. Invoice table created a record for a successfully placed order:

	OR...	INVOICEID	INVOICE_TYPE	INVOICEAMOUNT
1	999	I123	D	5100

There are certain unit test cases that we created to test the functionality such as:

1. *Price Engine*: A procedure to check the total price of all products added in the cart

2. *Quantity in shopping cart*: A procedure that returns the quantity of products in shopping cart.

3. *Products in shopping cart*: A procedure that returns the product ID in shopping cart.

SPECIAL ACKNOWLEDGEMENT

At last, we would like to thank Professor Nurcan Yuruk. It was really exciting to work on such a thought-provoking project. We used all the concepts that we learned in our database design course thus making it more fruitful. After completing this project, we feel that we are on a good foundation on how to use our gained knowledge into designing a real-time database system.