GAMING ARENA MANAGEMENT SYSTEM

Joshua Fernandes(22169738)

Muhammad Salman(21173110)

Module Co-ordinator: Parnia Samimi

Module Title: Advanced Databases

Module Code:CMP7214

BIRMINGHAM CITY
University

Table of Contents

1.Domain Description

This is a Gaming Store which has a gaming arena full of PCs,PS5s and many other consoles located inside it.This Gaming store has many different departments (id,Name,Type,is_manager,is_active,created_on,updated_on) who employ staff(ID,department_id, first_name, last_name, DOB, Salary, is_manager,is_active, created_on, updated_on). The department sells products(id,name,expiry,refundable,quantity) in the gaming store. In addition to that the department will provide services(id ,type ,charges ,playing_time_in_Hours , is_active , created_on ,updated_on) in the gaming arena. The customer(id,fname,lname,email,phone_no, Address ,shipping_Address ,billing_address ,created_on,updated_on) has the option to purchase a product, a service or both.The department processes this transaction (id,customer_id,staff_id,department_id,date ,time,Amount,Points_Received,ext_trans_token, Products,services).Each customer has the option to purchase one or many member cards(id,type,Date_Of_Purchase,amount,customer_id,is_active,created_on,updated_on) which will enable you to get discounts(member_card_id,serial_no,discount_amount,product_id,points,is_active,created_on,updated_on) on various products. The staff must also maintain the equipment(ID,Name, application, brand,quantity). Customers should be facilitated by the department.

2.Database Analysis

2.1 Business Rules

1. Every customer must have a phone number or email address.
2. Each Customer can only have one discount applied to their transactions.
3. Each customer can only use the gaming arena for a maximum of 4 hours at a time.
4. Department can employ many staff but one staff can only work for one department
5. Departments can only facilitate customers who are buying a product being sold by their own department.
6. The system must keep track of each transaction whether it be regarding the gaming store or gaming arena
7. A customer can buy many games at a time.
8. Games can be refunded up to 14 days after purchase.
9. Purchase of a membership card gives you a 20% discount.
10. Each Department must have a manager.
11. Each Department can only have one manager
12. Only Debit cards will be accepted not credit cards

2.2 List of Entities/Attributes

B_Cards

- Code(PK)

- max_amount
- Min_amount
- type
- is_active
- created_at
- updated_at
-

Department

- id (PK)
- name
- type
- is_active
- created_at
- updated_at

Customer

- id(PK)
- first_name
- last_name
- email
- phone_no
- dob
- Created_at
- Updated_at

Staff

- Id (PK)
- department_id(FK)
- first_name
- last_name
- dob
- salary
- is_active
- created_at
- updated_at

Customer Addresses

- id(PK)
- customer_id(FK)
- city
- state
- post_code

- country
- type
- is_active
- created_at
- updated_at

Staff Addresses

- id(PK)
- staff_id(FK)
- city
- state
- post_code
- country
- type
- is_active
- created_at
- updated_at

Equipment

- id(PK)
- department_id(FK)
- name
- brand
- application
- quantity

Member Card

- Id (PK)
- customer_id
- member_card_code
- amount
- type
- date_of_Purchase
- is_active
- created_at
- updated_at

Transaction

- id(PK)
- customer_id (FK)
- order_id
- staff_id (FK)
- department_id (FK)
- member_card_id (FK)
- Points_received
- date
- time
- Ext_trans_token

Services

- Id(PK)
- Department_id
- Type
- playing_time
- charges
- is_active
- created_at
- updated_at

Products

- id (PK)
- departmentid(FK)
- name
- type
- brand
- expiry
- refundable
- quantity
- created_at
- updated_at

Discount

- serial_no
- membercard_code(FK)
- product_id (FK)
- discount_amount
- is_active
- created_on
- updated_on

Order

- id(PK)
- order_at
- order_expiry
- customer_id
- status
- comments
- total_amount

Manager

- id(PK)
- department_id(FK)
- staff_id(FK)
- is_active

Order Details

- id(PK)
- order_id(FK)
- department_id(FK)

- product_id(FK)
- staff_id(FK)
- service_id(FK)
- services_hours
- product_quantity
- amount

2.3 Relationships

1:1 Relationships

[DEPARTMENT] 1<PROVIDES>1 [SERVICE]

[MANAGER] 1 <MANAGES>1 [DEPARTMENT]

[DEPARTMENT] 1<EMPLOYS> 1[MANAGER]

[STAFF LEAD] 1 <MANAGES> 1 [DEPARTMENT]

[DEPARTMENT] 1<EMPLOYS>1 [STAFF LEAD]

1:M Relationships

[DEPARTMENT] 1<EMPLOYS> M[STAFF]

[DEPARTMENT] 1 <SELLS> M [PRODUCTS]

[DEPARTMENT] 1<PROCESSES>M [TRANSACTION]

[CUSTOMER] 1<MAKES> M [TRANSACTION]

[CUSTOMER] 1<PURCHASES>M [MEMBER CARD]

[CUSTOMER] 1 <PURCHASES> M [PRODUCTS]

[CUSTOMER] 1 <PURCHASES> M [SERVICES]

[MEMBERCARD] 1 <HAS> M [DISCOUNT]

[PRODUCTS]1<HAS> M[DISCOUNT]

[DEPARTMENT] 1<FACILITATES> M[CUSTOMER]

M:N Relationships

[DEPARTMENT] M <MAINTAINS> N[EQUIPMENT]

[DEPARTMENT] M<FACILITATES> N[CUSTOMER]

2.4 Connectivities,Cardinality and Participation

A Department employs a minimum of 1 staff

A Department employs a maximum of M staff

A Staff is employed by a minimum of 1 department

A Staff is employed by a maximum of 1 department

A Department has a minimum of 1 manager

A Department has a maximum of 1 manager

A manager can manage a minimum of 1 department

A manager can manage a maximum of 1 department

A Department has a minimum of 1 staff lead

A Department has a maximum of 1 staff lead

A staff lead can lead a minimum of 1 department

A staff lead can lead a maximum of 1 department

A Department sells a minimum of 1 product
A Department sells a maximum of M products
A product is sold by a minimum of 1 department
A product is sold by a maximum of 1 department

A Customer can purchase a minimum of 1 service
A Customer can purchase a maximum of 1 service
Each Service can be used by a minimum of 1 person
Each Service can be used by a maximum of 1 person

Department provides a minimum of 1 service
Department provides a maximum of 1 service
A service can be provided by a minimum of 1 department
A service can be provided by a maximum of 1 department

A Customer can purchase a minimum of 1 product
A Customer can purchase a maximum of M Products
A Product can be purchased by a minimum of 1 customer
A Product can be purchased by a maximum of M customers

A product can have a minimum of 1 discount
A product can have a maximum of M discount
A discount can be placed on a minimum of 1 product
A discount can be placed on a maximum of 1 product

A department can facilitate a minimum of 1 customer
A department can facilitate a maximum of M customers
A customer can be facilitated by a minimum of 1 department
A customer can be facilitated by a maximum of 1 department

A customer can purchase a minimum of 1 membership card
A customer can purchase a maximum of M membership cards
A membership card can be purchased by a minimum of 1 customer
A membership card can be purchased by a maximum of M customers

A Customer can make a minimum of 1 transaction
A Customer can make a maximum of M transactions
A transaction can be made by a minimum of 1 customer
A transaction can be made by a maximum of M customers

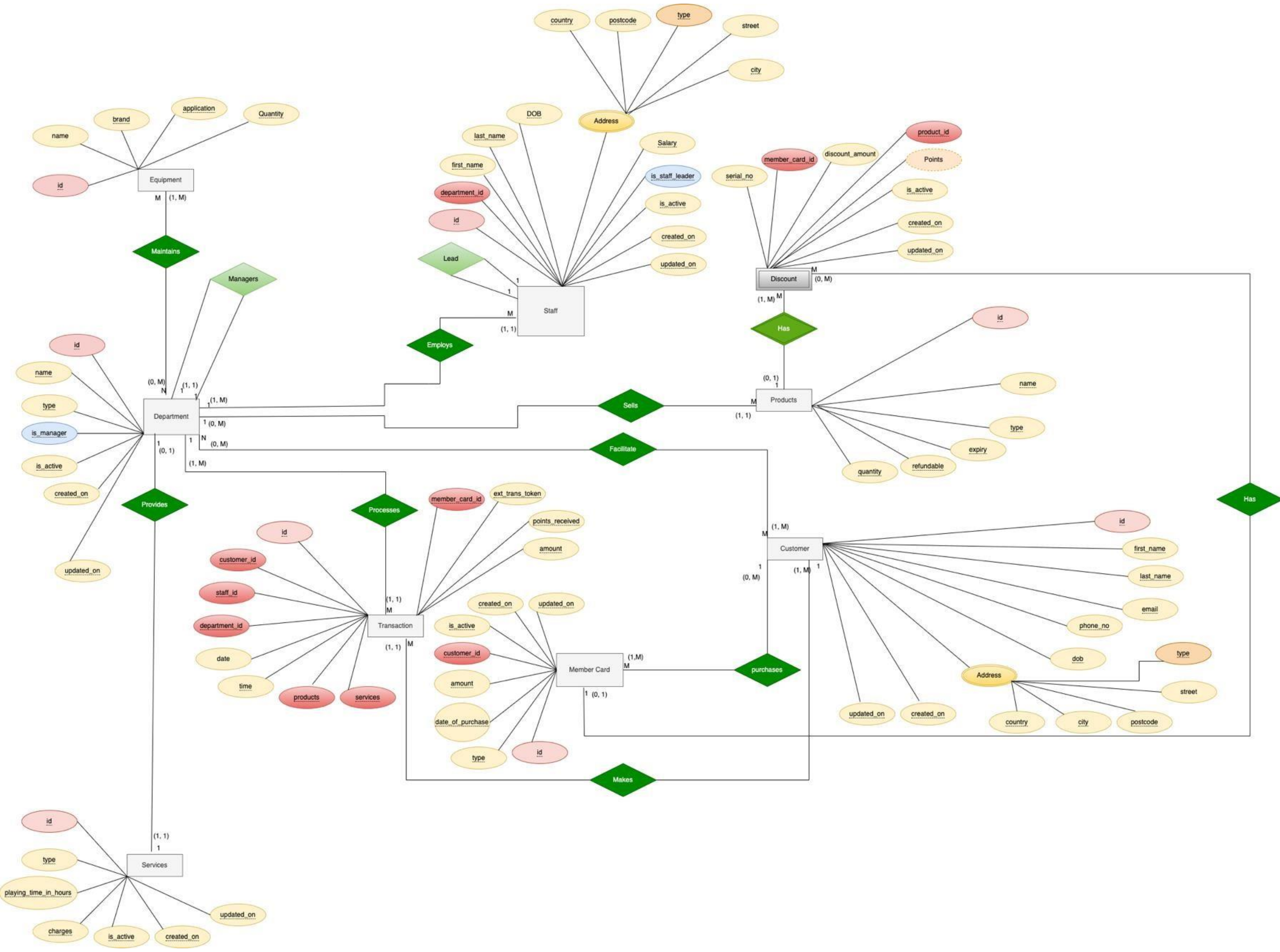
A department can process a minimum of 1 transaction

A department can process a maximum of M transactions
A transaction can be processed by a minimum of 1 department
A transaction can be processed by a maximum of 1 department

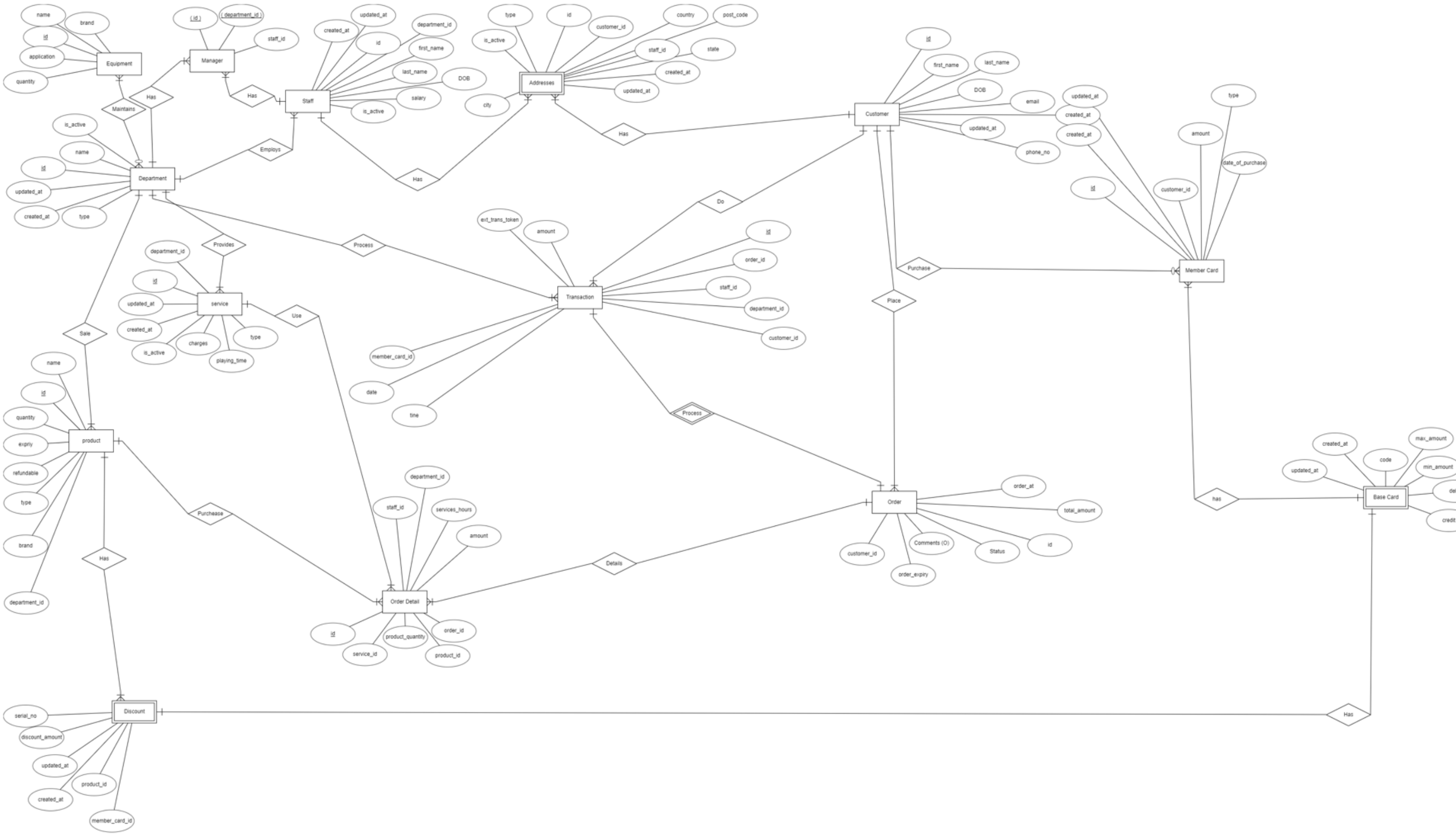
A Department maintains a minimum of 0 equipment
A Department maintains a maximum of M equipment
Equipment can be maintained by 1 department
Equipment can be maintained by M departments

3.Conceptual Model

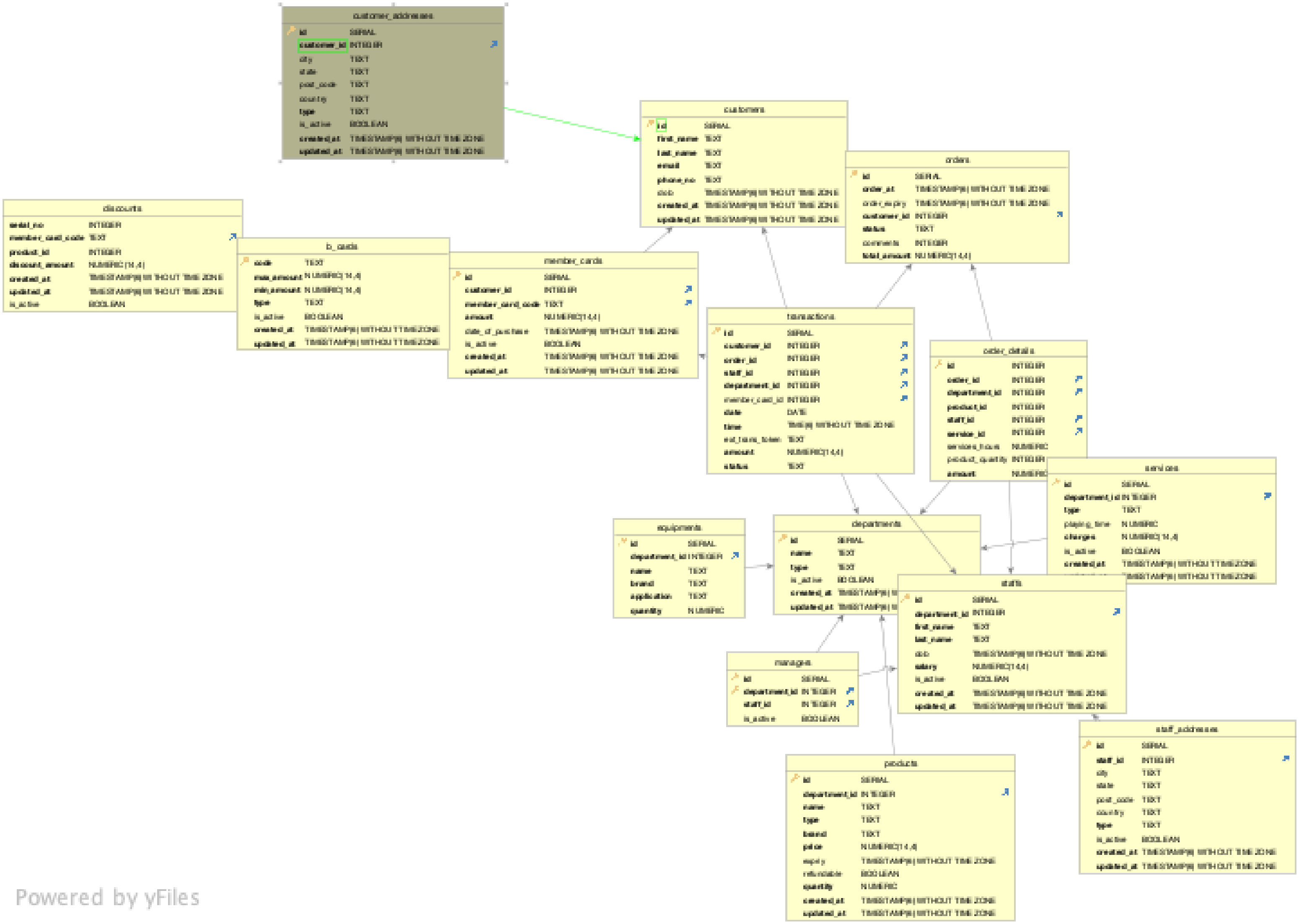
3.1 ERD



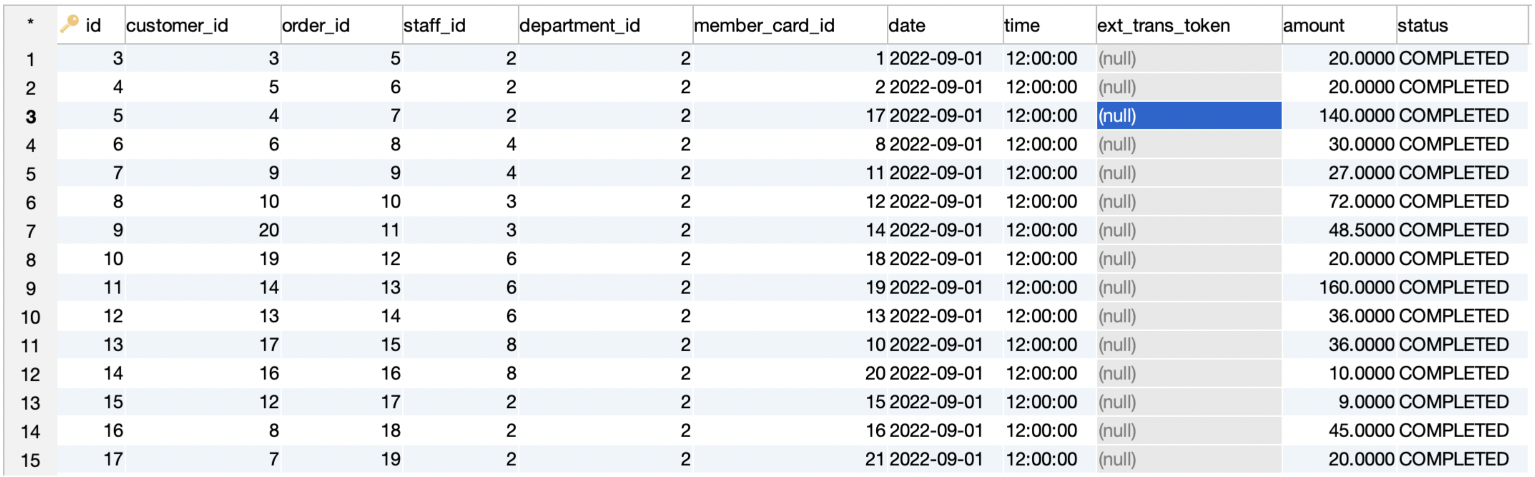
3.2 Extended Entity Relationship Diagram



3.3 Relationship Schema



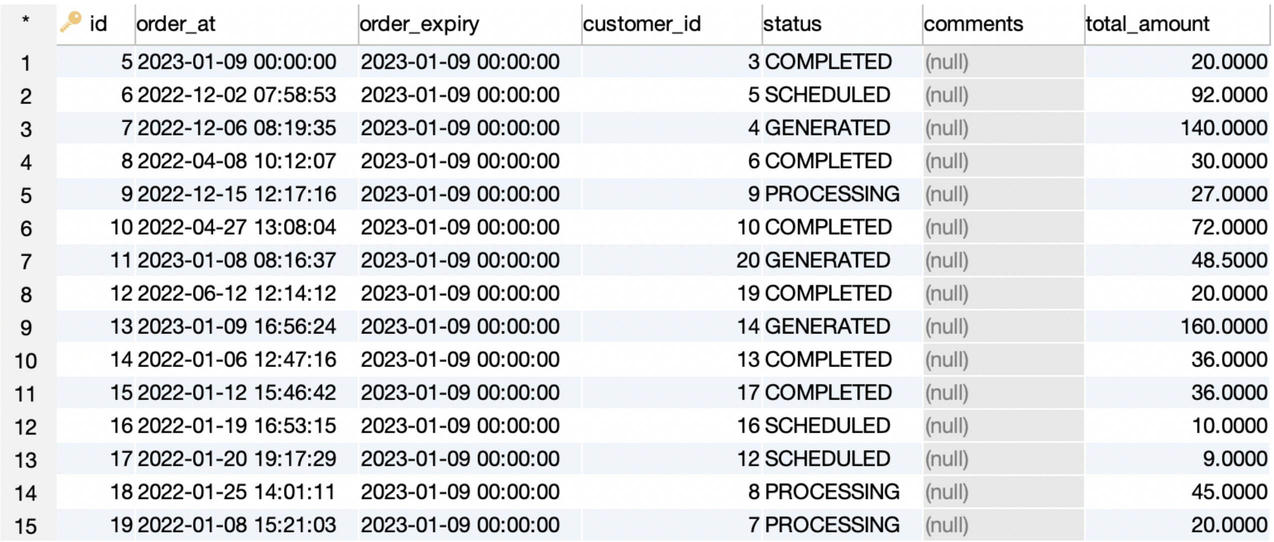
4.Normalisation



*	id	customer_id	order_id	staff_id	department_id	member_card_id	date	time	ext_trans_token	amount	status
1	3	3	5	2	2	1	2022-09-01	12:00:00	(null)	20.0000	COMPLETED
2	4	5	6	2	2	2	2022-09-01	12:00:00	(null)	20.0000	COMPLETED
3	5	4	7	2	2	17	2022-09-01	12:00:00	(null)	140.0000	COMPLETED
4	6	6	8	4	2	8	2022-09-01	12:00:00	(null)	30.0000	COMPLETED
5	7	9	9	4	2	11	2022-09-01	12:00:00	(null)	27.0000	COMPLETED
6	8	10	10	3	2	12	2022-09-01	12:00:00	(null)	72.0000	COMPLETED
7	9	20	11	3	2	14	2022-09-01	12:00:00	(null)	48.5000	COMPLETED
8	10	19	12	6	2	18	2022-09-01	12:00:00	(null)	20.0000	COMPLETED
9	11	14	13	6	2	19	2022-09-01	12:00:00	(null)	160.0000	COMPLETED
10	12	13	14	6	2	13	2022-09-01	12:00:00	(null)	36.0000	COMPLETED
11	13	17	15	8	2	10	2022-09-01	12:00:00	(null)	36.0000	COMPLETED
12	14	16	16	8	2	20	2022-09-01	12:00:00	(null)	10.0000	COMPLETED
13	15	12	17	2	2	15	2022-09-01	12:00:00	(null)	9.0000	COMPLETED
14	16	8	18	2	2	16	2022-09-01	12:00:00	(null)	45.0000	COMPLETED
15	17	7	19	2	2	21	2022-09-01	12:00:00	(null)	20.0000	COMPLETED

Figure 4 :Transaction Table

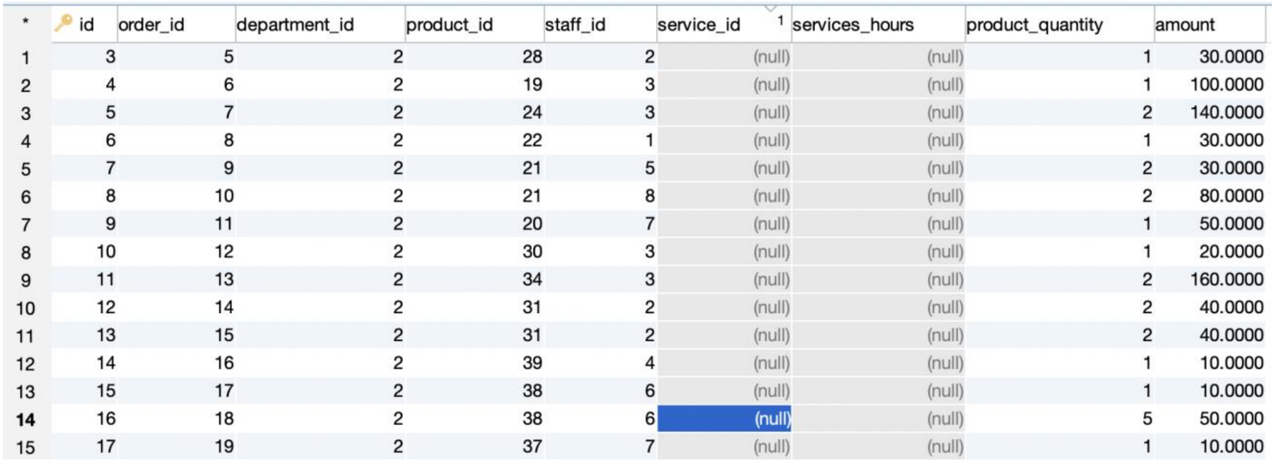
Figure 4 show the transaction table in it's normalised form.As there are no dependencies this is in the 3NF form. The primary key is id.The foreign keys are customer_id,order_id ,staff_id,department_id and member_card_id.



*	id	order_at	order_expiry	customer_id	status	comments	total_amount
1	5	2023-01-09 00:00:00	2023-01-09 00:00:00	3	COMPLETED	(null)	20.0000
2	6	2022-12-02 07:58:53	2023-01-09 00:00:00	5	SCHEDULED	(null)	92.0000
3	7	2022-12-06 08:19:35	2023-01-09 00:00:00	4	GENERATED	(null)	140.0000
4	8	2022-04-08 10:12:07	2023-01-09 00:00:00	6	COMPLETED	(null)	30.0000
5	9	2022-12-15 12:17:16	2023-01-09 00:00:00	9	PROCESSING	(null)	27.0000
6	10	2022-04-27 13:08:04	2023-01-09 00:00:00	10	COMPLETED	(null)	72.0000
7	11	2023-01-08 08:16:37	2023-01-09 00:00:00	20	GENERATED	(null)	48.5000
8	12	2022-06-12 12:14:12	2023-01-09 00:00:00	19	COMPLETED	(null)	20.0000
9	13	2023-01-09 16:56:24	2023-01-09 00:00:00	14	GENERATED	(null)	160.0000
10	14	2022-01-06 12:47:16	2023-01-09 00:00:00	13	COMPLETED	(null)	36.0000
11	15	2022-01-12 15:46:42	2023-01-09 00:00:00	17	COMPLETED	(null)	36.0000
12	16	2022-01-19 16:53:15	2023-01-09 00:00:00	16	SCHEDULED	(null)	10.0000
13	17	2022-01-20 19:17:29	2023-01-09 00:00:00	12	SCHEDULED	(null)	9.0000
14	18	2022-01-25 14:01:11	2023-01-09 00:00:00	8	PROCESSING	(null)	45.0000
15	19	2022-01-08 15:21:03	2023-01-09 00:00:00	7	PROCESSING	(null)	20.0000

Figure 5: Orders table

Figure 5 show the order table in its normalised form. As there are no transitive or functional dependencies this is in the 3NF form. The primary key is id. The foreign keys is customer_id.

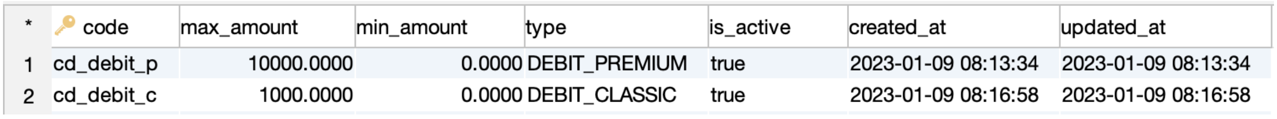


*	id	order_id	department_id	product_id	staff_id	service_id	services_hours	product_quantity	amount
1	3	5	2	28	2	(null)	(null)	1	30.0000
2	4	6	2	19	3	(null)	(null)	1	100.0000
3	5	7	2	24	3	(null)	(null)	2	140.0000
4	6	8	2	22	1	(null)	(null)	1	30.0000
5	7	9	2	21	5	(null)	(null)	2	30.0000
6	8	10	2	21	8	(null)	(null)	2	80.0000
7	9	11	2	20	7	(null)	(null)	1	50.0000
8	10	12	2	30	3	(null)	(null)	1	20.0000
9	11	13	2	34	3	(null)	(null)	2	160.0000
10	12	14	2	31	2	(null)	(null)	2	40.0000
11	13	15	2	31	2	(null)	(null)	2	40.0000
12	14	16	2	39	4	(null)	(null)	1	10.0000
13	15	17	2	38	6	(null)	(null)	1	10.0000
14	16	18	2	38	6	(null)	(null)	5	50.0000
15	17	19	2	37	7	(null)	(null)	1	10.0000

Figure 6: Order Details Table

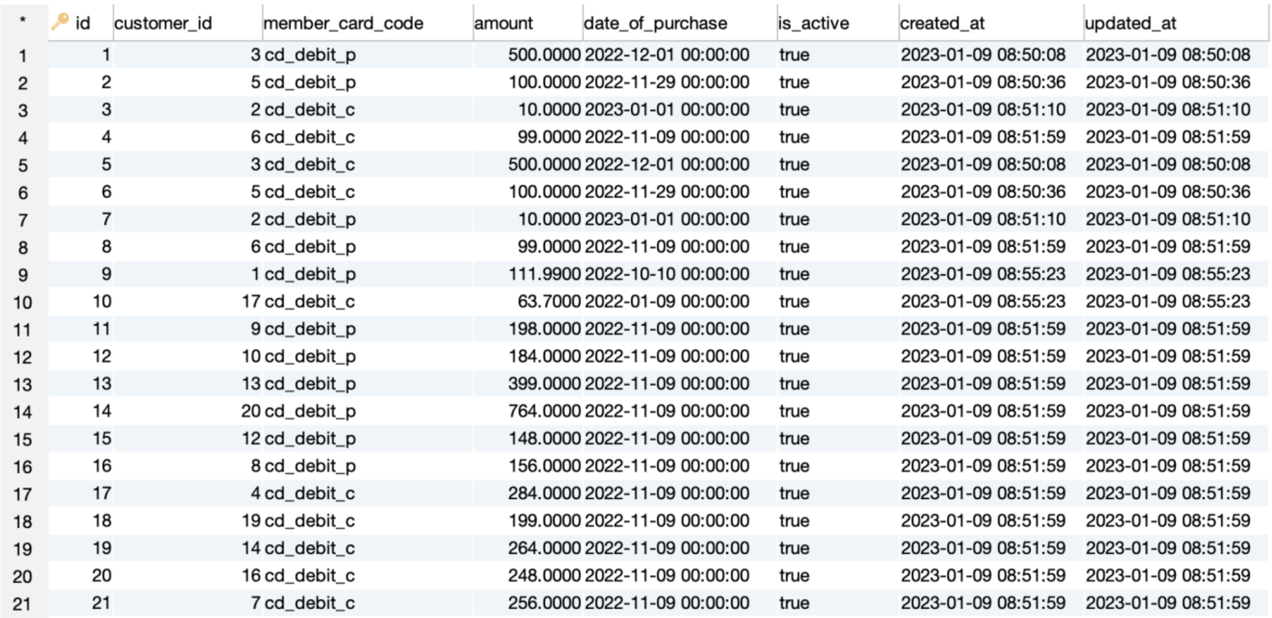
Figure 6 show the order table in its normalised form. As there are no transitive or functional dependencies this is in the 3NF form. The primary key is id. The foreign keys are order_id,department_id,product_id and staff_id.

When in 1NF both the order details and order tables would all be in 1 table but using foreign keys.



*	code	max_amount	min_amount	type	is_active	created_at	updated_at
1	cd_debit_p	10000.0000	0.0000	DEBIT_PREMIUM	true	2023-01-09 08:13:34	2023-01-09 08:13:34
2	cd_debit_c	1000.0000	0.0000	DEBIT_CLASSIC	true	2023-01-09 08:16:58	2023-01-09 08:16:58

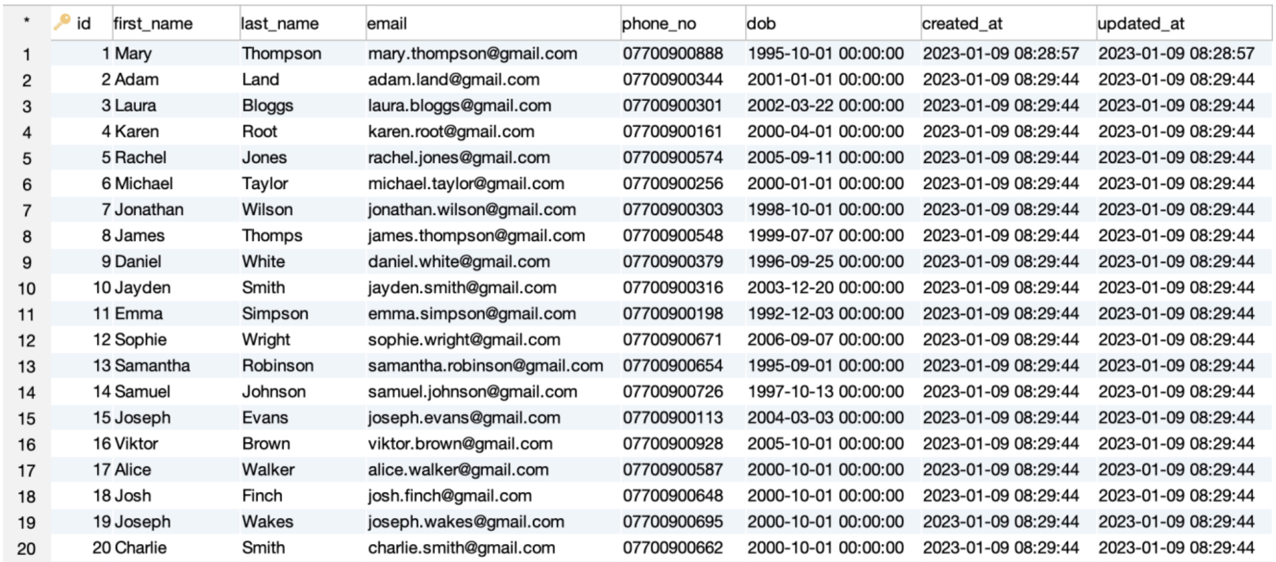
Figure 7:B_cards table



*	id	customer_id	member_card_code	amount	date_of_purchase	is_active	created_at	updated_at
1	1	3	cd_debit_p	500.0000	2022-12-01 00:00:00	true	2023-01-09 08:50:08	2023-01-09 08:50:08
2	2	5	cd_debit_p	100.0000	2022-11-29 00:00:00	true	2023-01-09 08:50:36	2023-01-09 08:50:36
3	3	2	cd_debit_c	10.0000	2023-01-01 00:00:00	true	2023-01-09 08:51:10	2023-01-09 08:51:10
4	4	6	cd_debit_c	99.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
5	5	3	cd_debit_c	500.0000	2022-12-01 00:00:00	true	2023-01-09 08:50:08	2023-01-09 08:50:08
6	6	5	cd_debit_c	100.0000	2022-11-29 00:00:00	true	2023-01-09 08:50:36	2023-01-09 08:50:36
7	7	2	cd_debit_p	10.0000	2023-01-01 00:00:00	true	2023-01-09 08:51:10	2023-01-09 08:51:10
8	8	6	cd_debit_p	99.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
9	9	1	cd_debit_p	111.9900	2022-10-10 00:00:00	true	2023-01-09 08:55:23	2023-01-09 08:55:23
10	10	17	cd_debit_c	63.7000	2022-01-09 00:00:00	true	2023-01-09 08:55:23	2023-01-09 08:55:23
11	11	9	cd_debit_p	198.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
12	12	10	cd_debit_p	184.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
13	13	13	cd_debit_p	399.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
14	14	20	cd_debit_p	764.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
15	15	12	cd_debit_p	148.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
16	16	8	cd_debit_p	156.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
17	17	4	cd_debit_c	284.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
18	18	19	cd_debit_c	199.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
19	19	14	cd_debit_c	264.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
20	20	16	cd_debit_c	248.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59
21	21	7	cd_debit_c	256.0000	2022-11-09 00:00:00	true	2023-01-09 08:51:59	2023-01-09 08:51:59

Figure 8 Member Cards Table

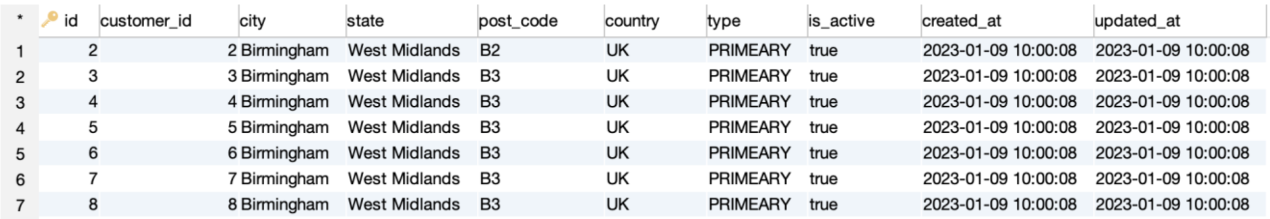
Figure 7 shows the Base cards table and Figure 8 shows the member tables. If left unnormalized these would both be in the same table. However since the tables have been split up this shows that this is in 3NF as neither tables have any transitive or functional dependencies. The primary key for the base cards table is code and the primary key for the member cards table is id.The base cards table has no foreign keys whereas the member cards table has customer_id as a foreign key.



*	id	first_name	last_name	email	phone_no	dob	created_at	updated_at
1	1	Mary	Thompson	mary.thompson@gmail.com	07700900888	1995-10-01 00:00:00	2023-01-09 08:28:57	2023-01-09 08:28:57
2	2	Adam	Land	adam.land@gmail.com	07700900344	2001-01-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
3	3	Laura	Bloggs	laura.bloggs@gmail.com	07700900301	2002-03-22 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
4	4	Karen	Root	karen.root@gmail.com	07700900161	2000-04-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
5	5	Rachel	Jones	rachel.jones@gmail.com	07700900574	2005-09-11 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
6	6	Michael	Taylor	michael.taylor@gmail.com	07700900256	2000-01-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
7	7	Jonathan	Wilson	jonathan.wilson@gmail.com	07700900303	1998-10-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
8	8	James	Thomps	james.thompson@gmail.com	07700900548	1999-07-07 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
9	9	Daniel	White	daniel.white@gmail.com	07700900379	1996-09-25 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
10	10	Jayden	Smith	jayden.smith@gmail.com	07700900316	2003-12-20 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
11	11	Emma	Simpson	emma.simpson@gmail.com	07700900198	1992-12-03 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
12	12	Sophie	Wright	sophie.wright@gmail.com	07700900671	2006-09-07 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
13	13	Samantha	Robinson	samantha.robinson@gmail.com	07700900654	1995-09-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
14	14	Samuel	Johnson	samuel.johnson@gmail.com	07700900726	1997-10-13 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
15	15	Joseph	Evans	joseph.evans@gmail.com	07700900113	2004-03-03 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
16	16	Viktor	Brown	viktor.brown@gmail.com	07700900928	2005-10-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
17	17	Alice	Walker	alice.walker@gmail.com	07700900587	2000-10-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
18	18	Josh	Finch	josh.finch@gmail.com	07700900648	2000-10-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
19	19	Joseph	Wakes	joseph.wakes@gmail.com	07700900695	2000-10-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44
20	20	Charlie	Smith	charlie.smith@gmail.com	07700900662	2000-10-01 00:00:00	2023-01-09 08:29:44	2023-01-09 08:29:44

Figure 9 : Customer Table

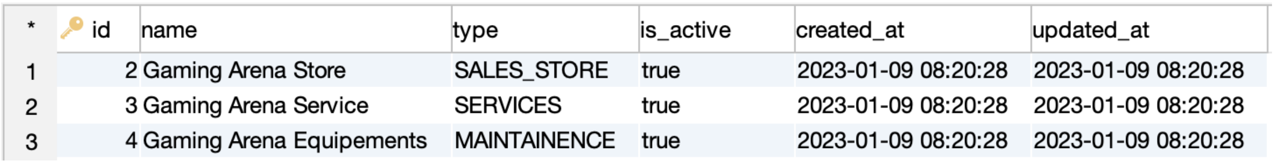
Figure 9 shows the customer table in the 3NF form as it has no functional or transitive dependencies.The primary key is id. The customer table does not have any foreign keys.



*	id	customer_id	city	state	post_code	country	type	is_active	created_at	updated_at
1	2	2	Birmingham	West Midlands	B2	UK	PRIMEARY	true	2023-01-09 10:00:08	2023-01-09 10:00:08
2	3	3	Birmingham	West Midlands	B3	UK	PRIMEARY	true	2023-01-09 10:00:08	2023-01-09 10:00:08
3	4	4	Birmingham	West Midlands	B3	UK	PRIMEARY	true	2023-01-09 10:00:08	2023-01-09 10:00:08
4	5	5	Birmingham	West Midlands	B3	UK	PRIMEARY	true	2023-01-09 10:00:08	2023-01-09 10:00:08
5	6	6	Birmingham	West Midlands	B3	UK	PRIMEARY	true	2023-01-09 10:00:08	2023-01-09 10:00:08
6	7	7	Birmingham	West Midlands	B3	UK	PRIMEARY	true	2023-01-09 10:00:08	2023-01-09 10:00:08
7	8	8	Birmingham	West Midlands	B3	UK	PRIMEARY	true	2023-01-09 10:00:08	2023-01-09 10:00:08

Figure 10: Customer Address Table

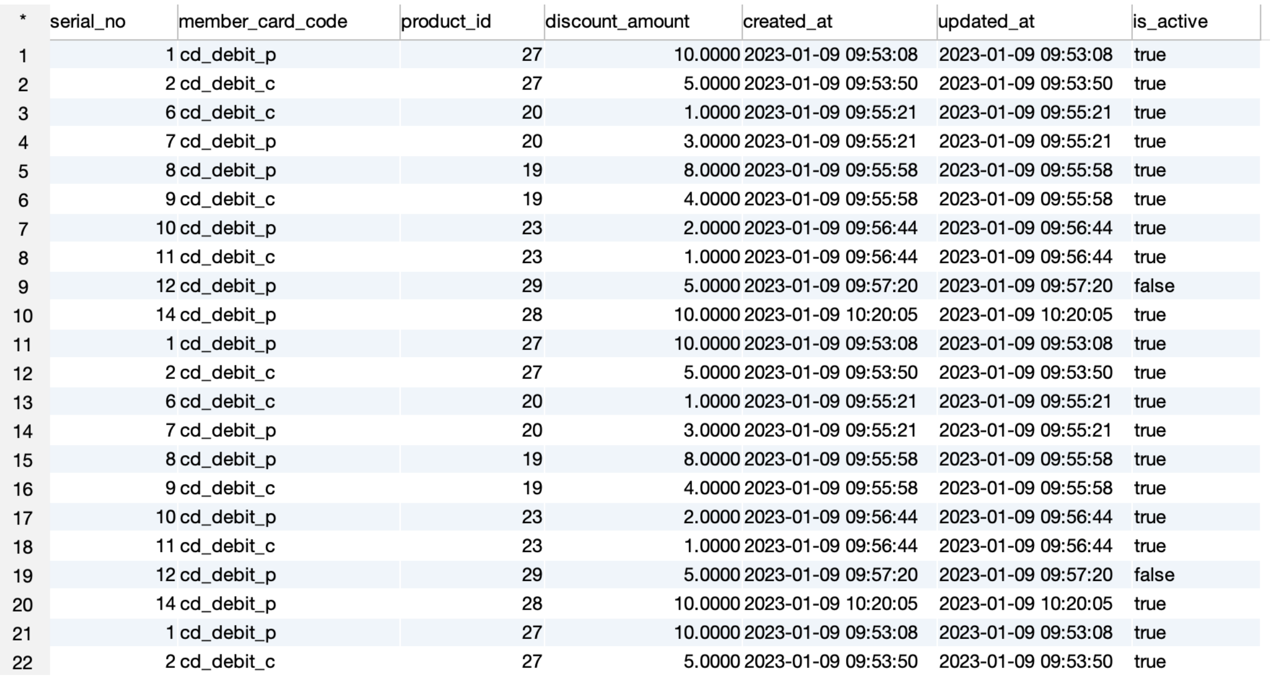
Figure 10 shows the customer address table in the 3NF form as it has no functional or transitive dependencies.The primary key is id. The customer address table does not have any foreign keys.



*	id	name	type	is_active	created_at	updated_at
1	2	Gaming Arena Store	SALES_STORE	true	2023-01-09 08:20:28	2023-01-09 08:20:28
2	3	Gaming Arena Service	SERVICES	true	2023-01-09 08:20:28	2023-01-09 08:20:28
3	4	Gaming Arena Equipements	MAINTAINENCE	true	2023-01-09 08:20:28	2023-01-09 08:20:28

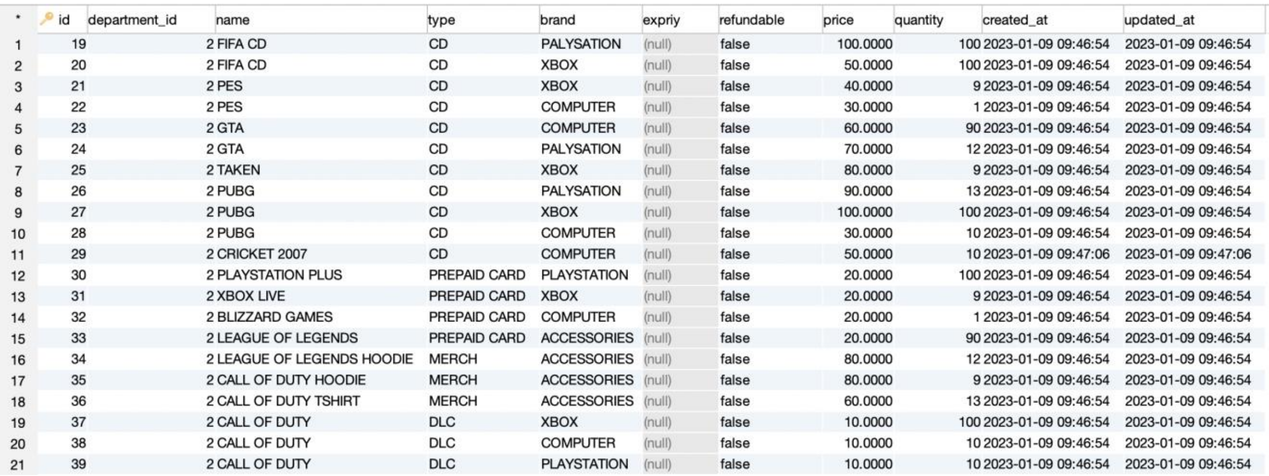
Figure 11: Department Table

Figure 11 shows the department table in the 3NF form as it has no functional or transitive dependencies.The primary key is id. The department table does not have any foreign keys.



*	serial_no	member_card_code	product_id	discount_amount	created_at	updated_at	is_active
1	1	cd_debit_p	27	10.0000	2023-01-09 09:53:08	2023-01-09 09:53:08	true
2	2	cd_debit_c	27	5.0000	2023-01-09 09:53:50	2023-01-09 09:53:50	true
3	6	cd_debit_c	20	1.0000	2023-01-09 09:55:21	2023-01-09 09:55:21	true
4	7	cd_debit_p	20	3.0000	2023-01-09 09:55:21	2023-01-09 09:55:21	true
5	8	cd_debit_p	19	8.0000	2023-01-09 09:55:58	2023-01-09 09:55:58	true
6	9	cd_debit_c	19	4.0000	2023-01-09 09:55:58	2023-01-09 09:55:58	true
7	10	cd_debit_p	23	2.0000	2023-01-09 09:56:44	2023-01-09 09:56:44	true
8	11	cd_debit_c	23	1.0000	2023-01-09 09:56:44	2023-01-09 09:56:44	true
9	12	cd_debit_p	29	5.0000	2023-01-09 09:57:20	2023-01-09 09:57:20	false
10	14	cd_debit_p	28	10.0000	2023-01-09 10:20:05	2023-01-09 10:20:05	true
11	1	cd_debit_p	27	10.0000	2023-01-09 09:53:08	2023-01-09 09:53:08	true
12	2	cd_debit_c	27	5.0000	2023-01-09 09:53:50	2023-01-09 09:53:50	true
13	6	cd_debit_c	20	1.0000	2023-01-09 09:55:21	2023-01-09 09:55:21	true
14	7	cd_debit_p	20	3.0000	2023-01-09 09:55:21	2023-01-09 09:55:21	true
15	8	cd_debit_p	19	8.0000	2023-01-09 09:55:58	2023-01-09 09:55:58	true
16	9	cd_debit_c	19	4.0000	2023-01-09 09:55:58	2023-01-09 09:55:58	true
17	10	cd_debit_p	23	2.0000	2023-01-09 09:56:44	2023-01-09 09:56:44	true
18	11	cd_debit_c	23	1.0000	2023-01-09 09:56:44	2023-01-09 09:56:44	true
19	12	cd_debit_p	29	5.0000	2023-01-09 09:57:20	2023-01-09 09:57:20	false
20	14	cd_debit_p	28	10.0000	2023-01-09 10:20:05	2023-01-09 10:20:05	true
21	1	cd_debit_p	27	10.0000	2023-01-09 09:53:08	2023-01-09 09:53:08	true
22	2	cd_debit_c	27	5.0000	2023-01-09 09:53:50	2023-01-09 09:53:50	true

Figure 12: Discounts Table



*	id	department_id	name	type	brand	expriy	refundable	price	quantity	created_at	updated_at
1	19	2	FIFA CD	CD	PALYSATION	(null)	false	100.0000	100	2023-01-09 09:46:54	2023-01-09 09:46:54
2	20	2	FIFA CD	CD	XBOX	(null)	false	50.0000	100	2023-01-09 09:46:54	2023-01-09 09:46:54
3	21	2	PES	CD	XBOX	(null)	false	40.0000	9	2023-01-09 09:46:54	2023-01-09 09:46:54
4	22	2	PES	CD	COMPUTER	(null)	false	30.0000	1	2023-01-09 09:46:54	2023-01-09 09:46:54
5	23	2	GTA	CD	COMPUTER	(null)	false	60.0000	90	2023-01-09 09:46:54	2023-01-09 09:46:54
6	24	2	GTA	CD	PALYSATION	(null)	false	70.0000	12	2023-01-09 09:46:54	2023-01-09 09:46:54
7	25	2	TAKEN	CD	XBOX	(null)	false	80.0000	9	2023-01-09 09:46:54	2023-01-09 09:46:54
8	26	2	PUBG	CD	PALYSATION	(null)	false	90.0000	13	2023-01-09 09:46:54	2023-01-09 09:46:54
9	27	2	PUBG	CD	XBOX	(null)	false	100.0000	100	2023-01-09 09:46:54	2023-01-09 09:46:54
10	28	2	PUBG	CD	COMPUTER	(null)	false	30.0000	10	2023-01-09 09:46:54	2023-01-09 09:46:54
11	29	2	CRICKET 2007	CD	COMPUTER	(null)	false	50.0000	10	2023-01-09 09:47:06	2023-01-09 09:47:06
12	30	2	PLAYSTATION PLUS	PREPAID CARD	PLAYSTATION	(null)	false	20.0000	100	2023-01-09 09:46:54	2023-01-09 09:46:54
13	31	2	XBOX LIVE	PREPAID CARD	XBOX	(null)	false	20.0000	9	2023-01-09 09:46:54	2023-01-09 09:46:54
14	32	2	BLIZZARD GAMES	PREPAID CARD	COMPUTER	(null)	false	20.0000	1	2023-01-09 09:46:54	2023-01-09 09:46:54
15	33	2	LEAGUE OF LEGENDS	PREPAID CARD	ACCESSORIES	(null)	false	20.0000	90	2023-01-09 09:46:54	2023-01-09 09:46:54
16	34	2	LEAGUE OF LEGENDS HOODIE	MERCH	ACCESSORIES	(null)	false	80.0000	12	2023-01-09 09:46:54	2023-01-09 09:46:54
17	35	2	CALL OF DUTY HOODIE	MERCH	ACCESSORIES	(null)	false	80.0000	9	2023-01-09 09:46:54	2023-01-09 09:46:54
18	36	2	CALL OF DUTY TSHIRT	MERCH	ACCESSORIES	(null)	false	60.0000	13	2023-01-09 09:46:54	2023-01-09 09:46:54
19	37	2	CALL OF DUTY	DLC	XBOX	(null)	false	10.0000	100	2023-01-09 09:46:54	2023-01-09 09:46:54
20	38	2	CALL OF DUTY	DLC	COMPUTER	(null)	false	10.0000	10	2023-01-09 09:46:54	2023-01-09 09:46:54
21	39	2	CALL OF DUTY	DLC	PLAYSTATION	(null)	false	10.0000	10	2023-01-09 09:46:54	2023-01-09 09:46:54

Figure 13:Products Table

Figure 12 shows the discounts table whereas figure 13 shows the products table. If the products were in the discount table this would be in the 1NF form as there would be a functional dependency between the product id and name there would also be repeated instances. When the tables are split there are no functional or transitive dependencies. The primary key for the products table is the id whereas the discounts table does not have a primary key. The foreign key for the discounts table is the product_id and the foreign key for the products table is department_id.

5.Restrictions

Creating Checks

```
ALTER TABLE b_cards
ADD CONSTRAINT amountcheck
CHECK (max_amount>min_amount);

ALTER TABLE customers
ADD CONSTRAINT phonenumbercheck
UNIQUE(phone_no);

ALTER TABLE customers
ADD CONSTRAINT emailcheck
UNIQUE(email);

ALTER TABLE departments
ADD CONSTRAINT dpt_typecheck
UNIQUE(type);

ALTER TABLE discounts
ADD CONSTRAINT discountmax
CHECK(discount_amount<50);

ALTER TABLE order_details
ADD CONSTRAINT serviceshoursnonzero
CHECK(services_hours>0);

ALTER TABLE order_details
ADD CONSTRAINT productqnonzero
CHECK(product_quantity>0);

ALTER TABLE order_details
ADD CONSTRAINT amountnonzero
CHECK(amount>0);

ALTER TABLE orders
ADD CONSTRAINT totalamountnonzero
CHECK(total_amount>0);

ALTER TABLE products
ADD CONSTRAINT quantitynonzero
CHECK(quantity>0);

ALTER TABLE services
ADD CONSTRAINT playingtimenotzero
CHECK(playing_time>0);
```

```
ALTER TABLE staffs
ADD CONSTRAINT salarynotzero
CHECK(salary>0);
```

5.Data Implementation

5.1Creating Database on Postgresql

Database: Postgresql

```
sudo su postgres
psql -c "CREATE USER test_user WITH PASSWORD 'test1234'"
psql -c 'DROP DATABASE "gaming_arena"'
psql -c 'CREATE DATABASE "gaming_arena"'
psql -c 'GRANT ALL PRIVILEGES ON DATABASE "gaming_arena" TO test_user'
exit
```

5.2Creating Tables

5.2.1 Creating base card table

```
CREATE TABLE b_cards
(
  code TEXT NOT NULL,
  max_amount NUMERIC(14,4) NOT NULL,
  min_amount NUMERIC(14,4) NOT NULL,
  type TEXT NOT NULL, -- DEBIT
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  PRIMARY KEY (code)
);
```

5.2.2 Creating Department Table

```
CREATE SEQUENCE departments_id_seq;
CREATE TABLE departments
(
  id INTEGER DEFAULT nextval('departments_id_seq'::regclass) NOT NULL,
  name TEXT NOT NULL,
  type TEXT NOT NULL,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  PRIMARY KEY (id)
);
```

5.2.3 Creating Customer Table

```
      CREATE SEQUENCE customers_id_seq;
CREATE TABLE customers
(
  id INTEGER DEFAULT nextval('customers_id_seq'::regclass) NOT NULL,
  first_name TEXT NOT NULL,
  last_name TEXT NOT NULL,
  email TEXT NOT NULL,
  phone_no TEXT NOT NULL,
  DOB TIMESTAMP(6) WITHOUT TIME ZONE,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  PRIMARY KEY (id)
);
```

5.2.4 Creating Staff table

```
CREATE SEQUENCE staffs_id_seq;
CREATE TABLE staffs
(
  id INTEGER DEFAULT nextval('staffs_id_seq'::regclass) NOT NULL,
  department_id INTEGER NOT NULL,
  first_name TEXT NOT NULL,
  last_name TEXT NOT NULL,
  DOB TIMESTAMP(6) WITHOUT TIME ZONE,
  salary NUMERIC(14,4) NOT NULL,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,

  PRIMARY KEY (id),
  CONSTRAINT fk_staff_department_id FOREIGN KEY
      (department_id) REFERENCES departments (id)
);
```

5.2.5 Creating staff Address table

```
CREATE SEQUENCE staff_addresses_id_seq;
CREATE TABLE staff_addresses
(
  id INTEGER DEFAULT nextval('staff_addresses_id_seq'::regclass) NOT NULL,
  staff_id INTEGER NOT NULL,
  city TEXT,
  state TEXT,
```

```sql
  post_code TEXT,
  country TEXT,
  type TEXT NOT NULL,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,

  PRIMARY KEY (id),
  CONSTRAINT fk_address_staff_id FOREIGN KEY
      (staff_id) REFERENCES staffs (id)

);
```

5.2.6 Creating Customer Address Table

```sql
CREATE SEQUENCE customer_addresses_id_seq;
CREATE TABLE customer_addresses
(
  id INTEGER DEFAULT nextval('customer_addresses_id_seq'::regclass) NOT NULL,
  customer_id INTEGER NOT NULL,
  city TEXT,
  state TEXT,
  post_code TEXT,
  country TEXT,
  type TEXT NOT NULL,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,

  PRIMARY KEY (id),
  CONSTRAINT fk_address_customer_id FOREIGN KEY
      (customer_id) REFERENCES customers (id)
);
```

5.2.7 Creating Products table

```sql
CREATE SEQUENCE products_id_seq;
CREATE TABLE products
(
  id INTEGER DEFAULT nextval('products_id_seq'::regclass) NOT NULL,
  department_id INTEGER NOT NULL,
  name TEXT NOT NULL,
  type TEXT NOT NULL,
  brand TEXT NOT NULL,
  expriy TIMESTAMP(6) WITHOUT TIME ZONE,
  refundable BOOLEAN DEFAULT true,
  quantity NUMERIC NOT NULL,
  price NUMERIC(14,4)NOT NULL,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT fk_product_department_id FOREIGN KEY
       (department_id) REFERENCES departments (id)
);
```

5.2.8 Creating Equipment table

```
CREATE SEQUENCE equipments_id_seq;
CREATE TABLE equipments
(
  id INTEGER DEFAULT nextval('equipments_id_seq'::regclass) NOT NULL,
  department_id INTEGER NOT NULL,
  name TEXT NOT NULL,
  brand TEXT NOT NULL,
  application TEXT NOT NULL,
  quantity NUMERIC NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT fk_equipements_department_id FOREIGN KEY
       (department_id) REFERENCES departments (id)
);
```

5.2.9 Creating Services Table

```
CREATE SEQUENCE services_id_seq;
CREATE TABLE services
(
  id INTEGER DEFAULT nextval('services_id_seq'::regclass) NOT NULL,
  department_id INTEGER NOT NULL,
  type TEXT NOT NULL,
  playing_time NUMERIC, --- In hours
  charges NUMERIC(14,4) NOT NULL,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,

  PRIMARY KEY (id),
  CONSTRAINT fk_services_department_id FOREIGN KEY
       (department_id) REFERENCES departments (id)
);
```

5.2.10 Creating Discount table

```
CREATE SEQUENCE discounts_id_seq;
CREATE TABLE discounts
(
  serial_no INTEGER DEFAULT nextval('discounts_id_seq'::regclass) NOT NULL,
  member_card_code TEXT NOT NULL,
  product_id INTEGER NOT NULL,
  discount_amount NUMERIC(14,4) NOT NULL,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  is_active BOOLEAN DEFAULT true,
  CONSTRAINT fk_discount_b_card_id FOREIGN KEY
       (member_card_code) REFERENCES b_cards (code),
  CONSTRAINT fk_discount_product_id FOREIGN KEY
       (product_id) REFERENCES products (id)
);
```

5.2.11 Creating Orders Table

```
CREATE SEQUENCE orders_id_seq;
CREATE TABLE orders
(
  id INTEGER DEFAULT nextval('orders_id_seq'::regclass) NOT NULL,
  order_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  order_expiry TIMESTAMP(6) WITHOUT TIME ZONE,
  customer_id  INTEGER NOT NULL,
  status TEXT NOT NULL, --- GENERATED, SCHEDULED, PROCESSING,
COMPLETED, CANCELLED
  Comments TEXT, ---- Cumstomer Notes
  total_amount NUMERIC(14,4) NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT fk_orders_customer_id FOREIGN KEY
       (customer_id) REFERENCES customers (id)
);
```

5.2.12 Creating Managers Table

```
CREATE SEQUENCE managers_id_seq;
CREATE TABLE managers
(
  id INTEGER DEFAULT nextval('managers_id_seq'::regclass) NOT NULL,
  department_id INTEGER NOT NULL,
  staff_id INTEGER NOT NULL,
  is_active BOOLEAN DEFAULT true,
  PRIMARY KEY(id, department_id),
  CONSTRAINT fk_manager_department_id FOREIGN KEY
      (department_id) REFERENCES departments (id),
  CONSTRAINT fk_managers_staff_id FOREIGN KEY
      (staff_id) REFERENCES staffs (id)
);
```

5.2.13 Creating Order Details Table

```
CREATE SEQUENCE oder_details_id_seq;
CREATE TABLE order_details
(
  id INTEGER DEFAULT nextval('oder_details_id_seq'::regclass) NOT NULL,
  order_id INTEGER NOT NULL,
  department_id INTEGER NOT NULL,
  product_id INTEGER,
  staff_id INTEGER NOT NULL,
  service_id INTEGER,
  services_hours NUMERIC,
  product_quantity INTEGER,
  amount NUMERIC(14,4) NOT NULL,
  PRIMARY KEY (id),

  CONSTRAINT fk_oder_details_oder_id FOREIGN KEY
      (order_id) REFERENCES orders (id),
  CONSTRAINT fk_oder_details_department_id FOREIGN KEY
      (department_id) REFERENCES departments (id),
  CONSTRAINT fk_order_details_product_id FOREIGN KEY
      (product_id) REFERENCES products (id),
  CONSTRAINT fk_order_details_service_id FOREIGN KEY
      (service_id) REFERENCES services (id),
  CONSTRAINT fk_order_details_staff_id FOREIGN KEY
      (staff_id) REFERENCES staffs (id)
);
```

5.2.14 Creating Members Cards table

```
CREATE SEQUENCE member_cards_id_seq;
CREATE TABLE member_cards
(
  id INTEGER DEFAULT nextval('member_cards_id_seq'::regclass) NOT NULL,
  customer_id INTEGER NOT NULL,
  member_card_code TEXT NOT NULL,
  amount NUMERIC(14,4) NOT NULL,
  date_of_purchase TIMESTAMP(6) WITHOUT TIME ZONE,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  updated_at TIMESTAMP(6) WITHOUT TIME ZONE DEFAULT now() NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT fk_memeber_cards_customer_id FOREIGN KEY
        (customer_id) REFERENCES customers (id),
  CONSTRAINT fk_memeber_cards_b_card_code FOREIGN KEY
        (member_card_code) REFERENCES b_cards (code)
);
```

5.2.15 Creating Transactions table

```
CREATE SEQUENCE transactions_id_seq;
CREATE TABLE Transactions
(
  id INTEGER DEFAULT nextval('transactions_id_seq'::regclass) NOT NULL,
  customer_id INTEGER NOT NULL,
  order_id INTEGER NOT NULL,
  staff_id INTEGER NOT NULL,
  department_id INTEGER NOT NULL,
  member_card_id INTEGER,
  date  DATE NOT NULL,
  time TIME(6) WITHOUT TIME ZONE NOT NULL,
  ext_trans_token TEXT, --- format: ext_xxxxxxxxxxxx
  amount NUMERIC(14,4) NOT NULL,
  status TEXT NOT NULL, --- PROCESSING, COMPLETED, FAILED, CANCELLED
  PRIMARY KEY (id),
  CONSTRAINT fk_trasactions_customer_id FOREIGN KEY
        (customer_id) REFERENCES customers (id),
  CONSTRAINT fk_trasactions_oder_id FOREIGN KEY
        (order_id) REFERENCES orders (id),
  CONSTRAINT fk_trasactions_staff_id FOREIGN KEY
        (staff_id) REFERENCES staffs (id),
  CONSTRAINT fk_transactions_department_id FOREIGN KEY
        (department_id) REFERENCES departments (id),
  CONSTRAINT fk_transactions_member_card_id FOREIGN KEY
        (member_card_id) REFERENCES member_cards (id)

);
```

5.3 Inserting Data

5.3.1 Inserting data into base card table

```
INSERT INTO b_cards (code, max_amount, min_amount, type, is_active, created_at,
updated_at) VALUES ('cd_debit_p', 10000.0000, 0.0000, 'DEBIT_PREMIUM', true, '2023-
01-09 08:13:34', '2023-01-09 08:13:34');
INSERT INTO b_cards (code, max_amount, min_amount, type, is_active, created_at,
updated_at) VALUES ('cd_debit_c', 1000.0000, 0.0000, 'DEBIT_CLASSIC', true, '2023-
01-09 08:16:58', '2023-01-09 08:16:58');
```

5.3.2 Inserting data into customers' table

```
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (1, 'Mary', 'Thompson', 'mary.thompson@gmail.com',
'07700900888 ', '1995-10-01 00:00:00', '2023-01-09 08:28:57', '2023-01-09 08:28:57');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (2, 'Adam', 'Land', 'adam.land@gmail.com', '07700900344', '2001-
01-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (3, 'Laura', 'Bloggs', 'laura.bloggs@gmail.com', '07700900301',
'2002-03-22 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (4, 'Karen', 'Root', 'karen.root@gmail.com', '07700900161', '2000-
04-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (5, 'Rachel', 'Jones', 'rachel.jones@gmail.com', '07700900574',
'2005-09-11 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (6, 'Michael', 'Taylor', 'michael.taylor@gmail.com', '07700900256',
'2000-01-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (7, 'Jonathan', 'Wilson', 'jonathan.wilson@gmail.com',
'07700900303', '1998-10-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (8, 'James', 'Thomps', 'james.thompson@gmail.com',
'07700900548 ', '1999-07-07 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (9, 'Daniel', 'White', 'daniel.white@gmail.com', '07700900379',
'1996-09-25 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (10, 'Jayden', 'Smith', 'jayden.smith@gmail.com', '07700900316',
'2003-12-20 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
```

```
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (11, 'Emma', 'Simpson', 'emma.simpson@gmail.com',
'07700900198', '1992-12-03 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (12, 'Sophie', 'Wright', 'sophie.wright@gmail.com', '07700900671',
'2006-09-07 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (13, 'Samantha', 'Robinson', 'samantha.robinson@gmail.com',
'07700900654', '1995-09-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (14, 'Samuel', 'Johnson', 'samuel.johnson@gmail.com',
'07700900726', '1997-10-13 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (15, 'Joseph', 'Evans', 'joseph.evans@gmail.com', '07700900113',
'2004-03-03 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (16, 'Viktor', 'Brown', 'viktor.brown@gmail.com', '07700900928 ',
'2005-10-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (17, 'Alice', 'Walker', 'alice.walker@gmail.com', '07700900587 ',
'2000-10-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (18, 'Josh', 'Finch', 'josh.finch@gmail.com', '07700900648 ', '2000-
10-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (19, 'Joseph', 'Wakes', 'joseph.wakes@gmail.com', '07700900695 ',
'2000-10-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at,
updated_at) VALUES (20, 'Charlie', 'Smith', 'charlie.smith@gmail.com', '07700900662 ',
'2000-10-01 00:00:00', '2023-01-09 08:29:44', '2023-01-09 08:29:44');
```

5.3.3 Inserting data into customer’s address table

```
INSERT INTO customer_addresses (id, customer_id, city, state, post_code, country, type,
is_active, created_at, updated_at) VALUES (2, 2, 'Birmingham', 'West Midlands', 'B2',
'UK', 'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO customer_addresses (id, customer_id, city, state, post_code, country, type,
is_active, created_at, updated_at) VALUES (3, 3, 'Birmingham', 'West Midlands', 'B3',
'UK', 'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO customer_addresses (id, customer_id, city, state, post_code, country, type,
is_active, created_at, updated_at) VALUES (4, 4, 'Birmingham', 'West Midlands', 'B3',
'UK', 'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO customer_addresses (id, customer_id, city, state, post_code, country, type,
is_active, created_at, updated_at) VALUES (5, 5, 'Birmingham', 'West Midlands', 'B3',
'UK', 'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO customer_addresses (id, customer_id, city, state, post_code, country, type,
is_active, created_at, updated_at) VALUES (6, 6, 'Birmingham', 'West Midlands', 'B3',
'UK', 'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO customer_addresses (id, customer_id, city, state, post_code, country, type,
is_active, created_at, updated_at) VALUES (7, 7, 'Birmingham', 'West Midlands', 'B3',
'UK', 'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO customer_addresses (id, customer_id, city, state, post_code, country, type,
is_active, created_at, updated_at) VALUES (8, 8, 'Birmingham', 'West Midlands', 'B3',
'UK', 'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
```

5.3.4 Inserting data into department table

```
INSERT INTO departments (id, name, type, is_active, created_at, updated_at) VALUES
(2, 'Gaming Arena Store', 'SALES_STORE', true, '2023-01-09 08:20:28', '2023-01-09
08:20:28');
INSERT INTO departments (id, name, type, is_active, created_at, updated_at) VALUES
(3, 'Gaming Arena Service', 'SERVICES', true, '2023-01-09 08:20:28', '2023-01-09
08:20:28');
INSERT INTO departments (id, name, type, is_active, created_at, updated_at) VALUES
(4, 'Gaming Arena Equipements', 'MAINTAINENCE', true, '2023-01-09 08:20:28', '2023-
01-09 08:20:28');
```

5.3.5 Inserting data into products table

```
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (19, 2, 'FIFA CD', 'CD', 'PALYSATION', null,
false, 100.0000, 100, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (20, 2, 'FIFA CD', 'CD', 'XBOX', null, false,
50.0000, 100, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (21, 2, 'PES', 'CD', 'XBOX', null, false,
40.0000, 9, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (22, 2, 'PES', 'CD', 'COMPUTER', null, false,
30.0000, 1, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (23, 2, 'GTA', 'CD', 'COMPUTER', null, false,
60.0000, 90, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (24, 2, 'GTA', 'CD', 'PALYSATION', null, false,
70.0000, 12, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (25, 2, 'TAKEN', 'CD', 'XBOX', null, false,
80.0000, 9, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (26, 2, 'PUBG', 'CD', 'PALYSATION', null,
false, 90.0000, 13, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (27, 2, 'PUBG', 'CD', 'XBOX', null, false,
100.0000, 100, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (28, 2, 'PUBG', 'CD', 'COMPUTER', null, false,
30.0000, 10, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (29, 2, 'CRICKET 2007', 'CD', 'COMPUTER',
null, false, 50.0000, 10, '2023-01-09 09:47:06', '2023-01-09 09:47:06');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (30, 2, 'PLAYSTATION PLUS', 'PREPAID
CARD', 'PLAYSTATION', null, false, 20.0000, 100, '2023-01-09 09:46:54', '2023-01-09
09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (31, 2, 'XBOX LIVE', 'PREPAID CARD',
'XBOX', null, false, 20.0000, 9, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price,
quantity, created_at, updated_at) VALUES (32, 2, 'BLIZZARD GAMES', 'PREPAID
CARD', 'COMPUTER', null, false, 20.0000, 1, '2023-01-09 09:46:54', '2023-01-09
09:46:54');
```

```
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price, quantity, created_at, updated_at) VALUES (33, 2, 'LEAGUE OF LEGENDS', 'PREPAID CARD', 'ACCESSORIES', null, false, 20.0000, 90, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price, quantity, created_at, updated_at) VALUES (34, 2, 'LEAGUE OF LEGENDS HOODIE', 'MERCH', 'ACCESSORIES', null, false, 80.0000, 12, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price, quantity, created_at, updated_at) VALUES (35, 2, 'CALL OF DUTY HOODIE', 'MERCH', 'ACCESSORIES', null, false, 80.0000, 9, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price, quantity, created_at, updated_at) VALUES (36, 2, 'CALL OF DUTY TSHIRT', 'MERCH', 'ACCESSORIES', null, false, 60.0000, 13, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price, quantity, created_at, updated_at) VALUES (37, 2, 'CALL OF DUTY', 'DLC', 'XBOX', null, false, 10.0000, 100, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price, quantity, created_at, updated_at) VALUES (38, 2, 'CALL OF DUTY', 'DLC', 'COMPUTER', null, false, 10.0000, 10, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
INSERT INTO products (id, department_id, name, type, brand, expriy, refundable, price, quantity, created_at, updated_at) VALUES (39, 2, 'CALL OF DUTY', 'DLC', 'PLAYSTATION', null, false, 10.0000, 10, '2023-01-09 09:46:54', '2023-01-09 09:46:54');
```

5.3.6 Inserting data into services table

```
INSERT INTO services (id, department_id, type, playing_time,charges,is_active,created_at,updated_at) VALUES (3, 2, 'PS5', 3,12,TRUE,'2022-04-09 08:12:34', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type, playing_time,charges,is_active,created_at,updated_at) VALUES (4, 2, 'PC', 4,16,TRUE,'2023-01-09 06:12:14', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type, playing_time,charges,is_active,created_at,updated_at) VALUES (5, 2, 'XBOX', 1,4,TRUE,'2023-01-09 07:20:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type, playing_time,charges,is_active,created_at,updated_at) VALUES (6, 2, 'SWITCH', 3,12,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type, playing_time,charges,is_active,created_at,updated_at) VALUES (7, 2, 'PC', 2,8,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type, playing_time,charges,is_active,created_at,updated_at) VALUES (8, 2, 'XBOX', 1,4,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
```

```sql
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (9, 2, 'PS4',
3,12,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (10, 2, 'PS5',
4,16,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (11, 2, 'PC',
3,12,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (12, 2, 'PC',
3,12,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (13, 2, 'PS5',
3,12,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (14, 2, 'XBOX',
1,4,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (15, 2, 'PS5',
4,16,TRUE,'2022-07-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (16, 2, 'PC',
3,12,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (17, 2, 'PS5',
2,8,TRUE,'2023-04-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (18, 2, 'SWITCH',
3,12,TRUE,'2022-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (19, 2, 'PS5',
1,4,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (20, 2, 'SWITCH',
3,12,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (21, 2, 'PS5',
2,8,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (22, 2, 'PC',
3,12,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO services (id, department_id, type,
playing_time,charges,is_active,created_at,updated_at) VALUES (23, 2, 'XBOX',
1,4,TRUE,'2023-01-09 08:41:53', '2023-01-09 08:41:53');
```

5.3.7 Inserting data into staff table

```
INSERT INTO staffs (id, department_id, first_name, last_name, dob, salary, is_active,
created_at, updated_at) VALUES (1, 2, 'ali', 'mujtaba', '1992-01-01 00:00:00',
10000.0000, true, '2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO staffs (id, department_id, first_name, last_name, dob, salary, is_active,
created_at, updated_at) VALUES (2, 2, 'zeeshan', 'shafiq', '1995-09-01 00:00:00',
5000.0000, true, '2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO staffs (id, department_id, first_name, last_name, dob, salary, is_active,
created_at, updated_at) VALUES (3, 2, 'ali', 'sajjad', '1999-11-25 00:00:00', 3000.0000,
true, '2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO staffs (id, department_id, first_name, last_name, dob, salary, is_active,
created_at, updated_at) VALUES (4, 3, 'Muhammad', 'Salman', '1998-01-09 00:00:00',
20000.0000, true, '2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO staffs (id, department_id, first_name, last_name, dob, salary, is_active,
created_at, updated_at) VALUES (5, 3, 'sajjel', 'hassan', '2000-01-09 00:00:00', 500.0000,
true, '2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO staffs (id, department_id, first_name, last_name, dob, salary, is_active,
created_at, updated_at) VALUES (6, 3, 'asad', 'Mehmood', '2002-01-09 00:00:00',
400.0000, true, '2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO staffs (id, department_id, first_name, last_name, dob, salary, is_active,
created_at, updated_at) VALUES (7, 4, 'John', 'cena', '1995-01-09 00:00:00', 1000.0000,
true, '2023-01-09 08:41:53', '2023-01-09 08:41:53');
INSERT INTO staffs (id, department_id, first_name, last_name, dob, salary, is_active,
created_at, updated_at) VALUES (8, 4, 'adam', 'kin', '1993-01-09 00:00:00', 500.0000,
true, '2023-01-09 08:41:53', '2023-01-09 08:41:53');
```

5.3.8 Inserting data into staff addresses table

```
INSERT INTO staff_addresses (id, staff_id, city, state, post_code, country, type, is_active,
created_at, updated_at) VALUES (2, 2, 'Birmingham', 'West Midlands', 'B2', 'UK',
'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO staff_addresses (id, staff_id, city, state, post_code, country, type, is_active,
created_at, updated_at) VALUES (3, 3, 'Birmingham', 'West Midlands', 'B3', 'UK',
'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO staff_addresses (id, staff_id, city, state, post_code, country, type, is_active,
created_at, updated_at) VALUES (4, 4, 'Birmingham', 'West Midlands', 'B3', 'UK',
'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO staff_addresses (id, staff_id, city, state, post_code, country, type, is_active,
created_at, updated_at) VALUES (5, 5, 'Birmingham', 'West Midlands', 'B3', 'UK',
'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO staff_addresses (id, staff_id, city, state, post_code, country, type, is_active,
created_at, updated_at) VALUES (6, 6, 'Birmingham', 'West Midlands', 'B3', 'UK',
'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
INSERT INTO staff_addresses (id, staff_id, city, state, post_code, country, type, is_active,
created_at, updated_at) VALUES (7, 7, 'Birmingham', 'West Midlands', 'B3', 'UK',
'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
```

```
INSERT INTO staff_addresses (id, staff_id, city, state, post_code, country, type, is_active,
created_at, updated_at) VALUES (8, 8, 'Birmingham', 'West Midlands', 'B3', 'UK',
'PRIMEARY', true, '2023-01-09 10:00:08', '2023-01-09 10:00:08');
```

5.3.9 Inserting data into Managers table

```
INSERT INTO managers (id, department_id, staff_id, is_active) VALUES (1, 2, 1, true);
INSERT INTO managers (id, department_id, staff_id, is_active) VALUES (2, 3, 4, true);
INSERT INTO managers (id, department_id, staff_id, is_active) VALUES (3, 4, 7, true);
```

5.3.10 Inserting data into Members card table

```
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (1, 3, 'cd_debit_p',
500.0000, '2022-12-01 00:00:00', true, '2023-01-09 08:50:08', '2023-01-09 08:50:08');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (2, 5, 'cd_debit_p',
100.0000, '2022-11-29 00:00:00', true, '2023-01-09 08:50:36', '2023-01-09 08:50:36');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (3, 2, 'cd_debit_c',
10.0000, '2023-01-01 00:00:00', true, '2023-01-09 08:51:10', '2023-01-09 08:51:10');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (4, 6, 'cd_debit_c',
99.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (5, 3, 'cd_debit_c',
500.0000, '2022-12-01 00:00:00', true, '2023-01-09 08:50:08', '2023-01-09 08:50:08');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (6, 5, 'cd_debit_c',
100.0000, '2022-11-29 00:00:00', true, '2023-01-09 08:50:36', '2023-01-09 08:50:36');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (7, 2, 'cd_debit_p',
10.0000, '2023-01-01 00:00:00', true, '2023-01-09 08:51:10', '2023-01-09 08:51:10');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (8, 6, 'cd_debit_p',
99.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (9, 1, 'cd_debit_p',
111.9900, '2022-10-10 00:00:00', true, '2023-01-09 08:55:23', '2023-01-09 08:55:23');
```

```
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (10, 17, 'cd_debit_c',
63.7000, '2022-01-09 00:00:00', true, '2023-01-09 08:55:23', '2023-01-09 08:55:23');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (11, 9, 'cd_debit_p',
198.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (12, 10, 'cd_debit_p',
184.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (13, 13, 'cd_debit_p',
399.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (14,20, 'cd_debit_p',
764.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (15, 12, 'cd_debit_p',
148.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (16, 8, 'cd_debit_p',
156.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (17, 4, 'cd_debit_c',
284.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (18, 19, 'cd_debit_c',
199.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (19,14, 'cd_debit_c',
264.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (20, 16, 'cd_debit_c',
248.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
INSERT INTO member_cards (id, customer_id, member_card_code, amount,
date_of_purchase, is_active, created_at, updated_at) VALUES (21, 7, 'cd_debit_c',
256.0000, '2022-11-09 00:00:00', true, '2023-01-09 08:51:59', '2023-01-09 08:51:59');
```

5.3.11 Inserting data into Discounts table

```
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (1, 'cd_debit_p', 27, 10.0000, '2023-01-09
09:53:08', '2023-01-09 09:53:08', true);
```

```
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (2, 'cd_debit_c', 27, 5.0000, '2023-01-09
09:53:50', '2023-01-09 09:53:50', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (6, 'cd_debit_c', 20, 1.0000, '2023-01-09
09:55:21', '2023-01-09 09:55:21', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (7, 'cd_debit_p', 20, 3.0000, '2023-01-09
09:55:21', '2023-01-09 09:55:21', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (8, 'cd_debit_p', 19, 8.0000, '2023-01-09
09:55:58', '2023-01-09 09:55:58', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (9, 'cd_debit_c', 19, 4.0000, '2023-01-09
09:55:58', '2023-01-09 09:55:58', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (10, 'cd_debit_p', 23, 2.0000, '2023-01-09
09:56:44', '2023-01-09 09:56:44', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (11, 'cd_debit_c', 23, 1.0000, '2023-01-09
09:56:44', '2023-01-09 09:56:44', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (12, 'cd_debit_p', 29, 5.0000, '2023-01-09
09:57:20', '2023-01-09 09:57:20', false);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (14, 'cd_debit_p', 28, 10.0000, '2023-01-09
10:20:05', '2023-01-09 10:20:05', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (15, 'cd_debit_p', 31, 10.0000, '2023-01-09
10:20:05', '2023-01-09 10:20:05', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (16, 'cd_debit_p', 38, 10.0000, '2023-01-09
10:20:05', '2023-01-09 10:20:05', true);
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount,
created_at, updated_at, is_active) VALUES (17, 'cd_debit_p', 21, 10.0000, '2023-01-09
10:20:05', '2023-01-09 10:20:05', true);
```

5.3.12 Inserting data into Equipment table

```sql
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(1, 4, 'PS5', 'SONY','PES',25);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(2, 4, 'PS5', 'SONY','Call of Duty',25);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(3, 4, 'PS5', 'SONY','PES',25);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(4, 4, 'PS5', 'SONY','Bloodborne',25);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(5, 4, 'XBOX', 'MICROSOFT','FIFA',25);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(6, 4, 'XBOX', 'MICROSOFT','Call of Duty',25);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(7, 4, 'XBOX', 'MICROSOFT','Gears of War',25);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(8, 4, 'XBOX', 'MICROSOFT','PES',25);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(9, 4, 'SWITCH', 'NINTENDO','Smash',5);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(10, 4, 'SWITCH', 'NINTENDO','Animal Crossing',5);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(11, 4, 'PC', 'RAZER','FIFA',35);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(12, 4, 'PC', 'RAZER','Valorant',35);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(13, 4, 'PC', 'RAZER','League of Legends',35);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(14, 4, 'PC', 'RAZER','Minecraft',35);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(15, 4, 'PC', 'RAZER','Overwatch',35);
INSERT INTO equipments (id, department_id, name, brand,application,quantity) VALUES
(16, 4, 'PC', 'RAZER','Steam',35);
```

5.3.13 Inserting data into Orders table

```
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (5, '2023-01-02 06:41:36', '2023-01-09 00:00:00', 3,
'COMPLETED', null, 20.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (6, '2022-12-02 07:58:53', '2023-01-09 00:00:00', 5,
'SCHEDULED', null, 92.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (7, '2022-12-06 08:19:35', '2023-01-09 00:00:00', 4,
'GENERATED', null, 140.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (8, '2022-04-08 10:12:07', '2023-01-09 00:00:00', 6,
'COMPLETED', null, 30.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (9, '2022-12-15 12:17:16', '2023-01-09 00:00:00', 9,
'PROCESSING', null, 27.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (10, '2022-04-27 13:08:04', '2023-01-09 00:00:00', 10,
'COMPLETED', null, 72.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (11, '2023-01-08 08:16:37', '2023-01-09 00:00:00', 20,
'GENERATED', null, 48.5000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (12, '2022-06-12 12:14:12', '2023-01-09 00:00:00', 19,
'COMPLETED', null, 20.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (13, '2023-01-09 16:56:24', '2023-01-09 00:00:00', 14,
'GENERATED', null, 160.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (14, '2022-01-06 12:47:16', '2023-01-09 00:00:00', 13,
'COMPLETED', null, 36.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (15, '2022-01-12 15:46:42', '2023-01-09 00:00:00', 17,
'COMPLETED', null, 36.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (16, '2022-01-19 16:53:15', '2023-01-09 00:00:00', 16,
'SCHEDULED', null, 10.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (17, '2022-01-20 19:17:29', '2023-01-09 00:00:00', 12,
'SCHEDULED', null, 9.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (18, '2022-01-25 14:01:11', '2023-01-09 00:00:00', 8,
'PROCESSING', null, 45.0000);
INSERT INTO orders (id, order_at, order_expiry, customer_id, status, comments,
total_amount) VALUES (19, '2022-01-08 15:21:03', '2023-01-09 00:00:00', 7,
'PROCESSING', null, 20.0000);
```

5.3.14 Inserting data into Order Details table

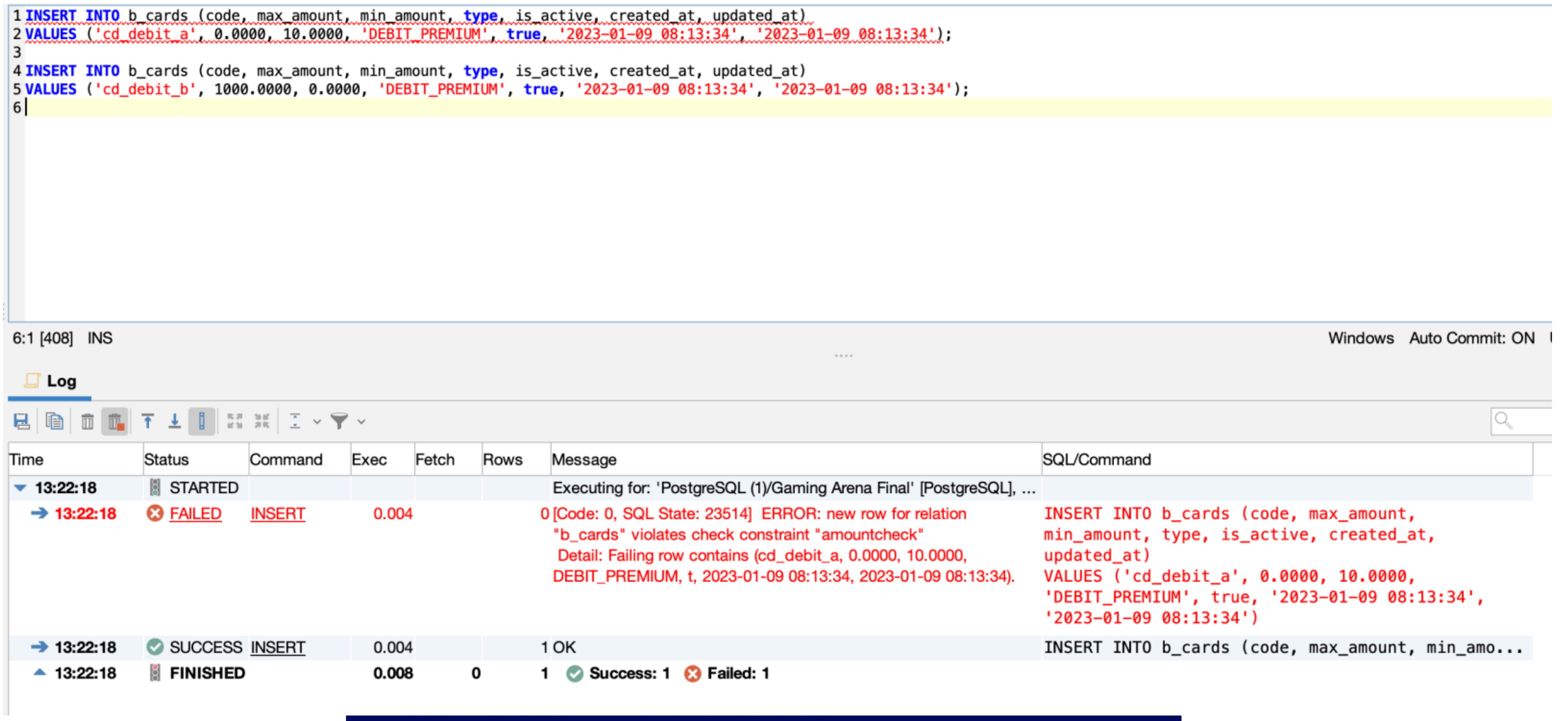
```
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (3, 5, 2, 28, 2, null, null, 1, 30.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (4, 6, 2, 19, 3, null, null, 1,
100.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (5, 7, 2, 24, 3, null, null, 2,
140.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (6, 8, 2, 22, 1, null, null, 1, 30.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (7, 9, 2, 21, 5, null, null, 2, 30.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (8, 10, 2, 21, 8, null, null, 2,
80.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (9, 11, 2, 20, 7, null, null, 1,
50.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (10, 12, 2, 30, 3, null, null, 1,
20.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (11, 13, 2, 34, 3, null, null, 2,
160.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (12, 14, 2, 31, 2, null, null, 2,
40.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (13, 15, 2, 31, 2, null, null, 2,
40.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (14, 16, 2, 39, 4, null, null, 1,
10.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (15, 17, 2, 38, 6, null, null, 1,
10.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (16, 18, 2, 38, 6, null, null, 5,
50.0000);
INSERT INTO order_details (id, order_id, department_id, product_id, staff_id, service_id,
services_hours, product_quantity, amount) VALUES (17, 19, 2, 37, 7, null, null, 1,
10.0000);
```

5.3.14 Inserting data into Transaction table

```
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (3, 3, 5, 2, 2, 1,
'2022-09-01', '12:00:00', null, 20.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (4, 5, 6, 2, 2, 2,
'2022-09-01', '12:00:00', null, 92.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (5, 4, 7, 2, 2, 17,
'2022-09-01', '12:00:00', null, 140.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (6, 6, 8, 4, 2, 8,
'2022-09-01', '12:00:00', null, 30.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (7, 9, 9, 4, 2, 11,
'2022-09-01', '12:00:00', null, 27.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (8, 10, 10, 3, 2,
12, '2022-09-01', '12:00:00', null, 72.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (9, 20, 11, 3, 2,
14, '2022-09-01', '12:00:00', null, 48.5000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (10, 19, 12, 6, 2,
18, '2022-09-01', '12:00:00', null, 20.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (11, 14, 13, 6, 2,
19, '2022-09-01', '12:00:00', null, 160.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (12, 13, 14, 6, 2,
13, '2022-09-01', '12:00:00', null, 36.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (13, 17, 15, 8, 2,
10, '2022-09-01', '12:00:00', null, 36.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (14, 16, 16, 8, 2,
20, '2022-09-01', '12:00:00', null, 10.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (15, 12, 17, 2, 2,
15, '2022-09-01', '12:00:00', null, 9.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (16, 8, 18, 2, 2,
16, '2022-09-01', '12:00:00', null, 45.0000, 'COMPLETED');
INSERT INTO transactions (id, customer_id, order_id, staff_id, department_id,
member_card_id, date, time, ext_trans_token, amount, status) VALUES (17, 7, 19, 2, 2,
21, '2022-09-01', '12:00:00', null, 20.0000, 'COMPLETED');
```

6.Tests

6.1Bcard Test



```sql
INSERT INTO b_cards (code, max_amount, min_amount, type, is_active, created_at, updated_at)
VALUES ('cd_debit_a', 0.0000, 10.0000, 'DEBIT_PREMIUM', true, '2023-01-09 08:13:34', '2023-01-09 08:13:34');

INSERT INTO b_cards (code, max_amount, min_amount, type, is_active, created_at, updated_at)
VALUES ('cd_debit_b', 1000.0000, 0.0000, 'DEBIT_PREMIUM', true, '2023-01-09 08:13:34', '2023-01-09 08:13:34');

```

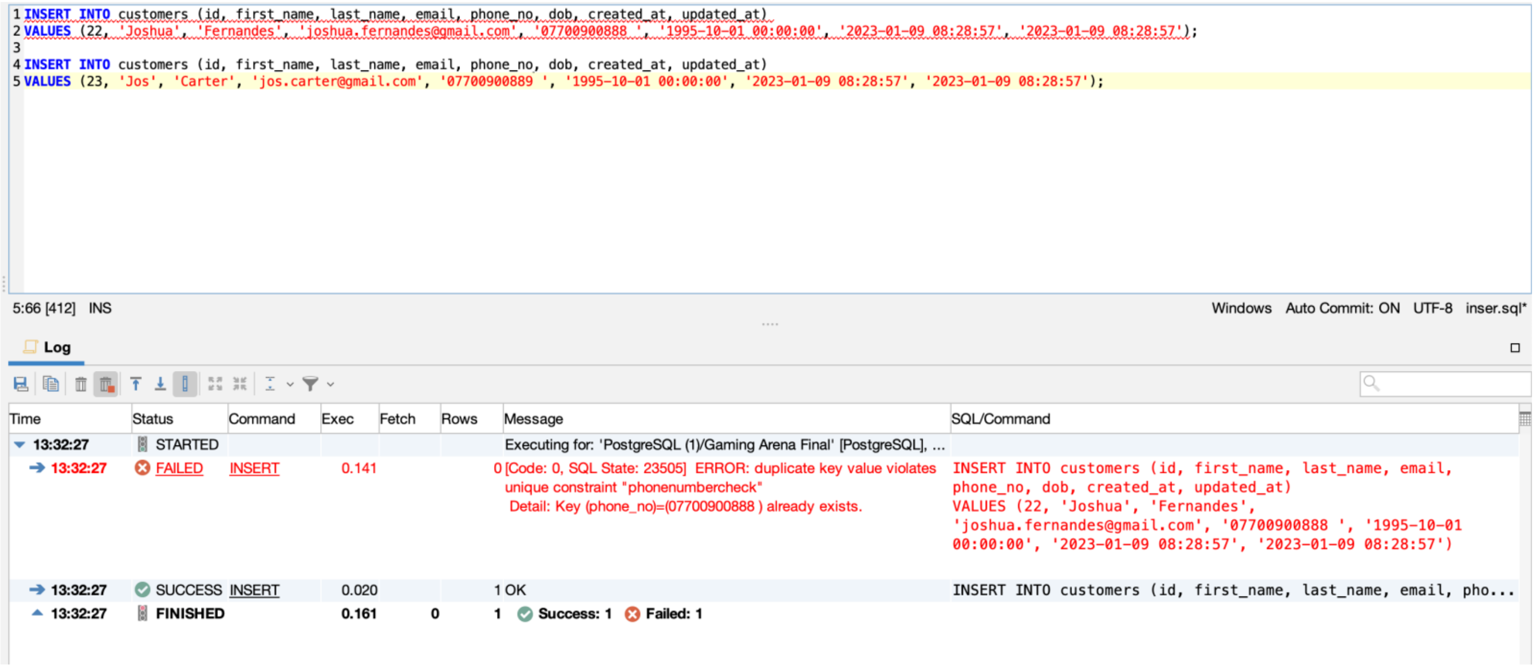
Time	Status	Command	Exec	Fetch	Rows	Message	SQL/Command
13:22:18	STARTED					Executing for: 'PostgreSQL (1)/Gaming Arena Final' [PostgreSQL], ...	

13:22:18	FAILED	INSERT	0.004		0	[Code: 0, SQL State: 23514] ERROR: new row for relation "b_cards" violates check constraint "amountcheck"
						Detail: Failing row contains (cd_debit_a, 0.0000, 10.0000, DEBIT_PREMIUM, t, 2023-01-09 08:13:34, 2023-01-09 08:13:34).
							INSERT INTO b_cards (code, max_amount, min_amount, type, is_active, created_at, updated_at)
							VALUES ('cd_debit_a', 0.0000, 10.0000, 'DEBIT_PREMIUM', true, '2023-01-09 08:13:34', '2023-01-09 08:13:34')

13:22:18	SUCCESS	INSERT	0.004		1	OK	INSERT INTO b_cards (code, max_amount, min_amo...
13:22:18	FINISHED		0.008	0	1	✓ Success: 1 ✗ Failed: 1	

This test shows that if max amount is less than min amount the row will not be added and error will be thrown. The second test shows that when the max amount is higher than the min amount the row will be added successfully.

6.2Phone Number Test

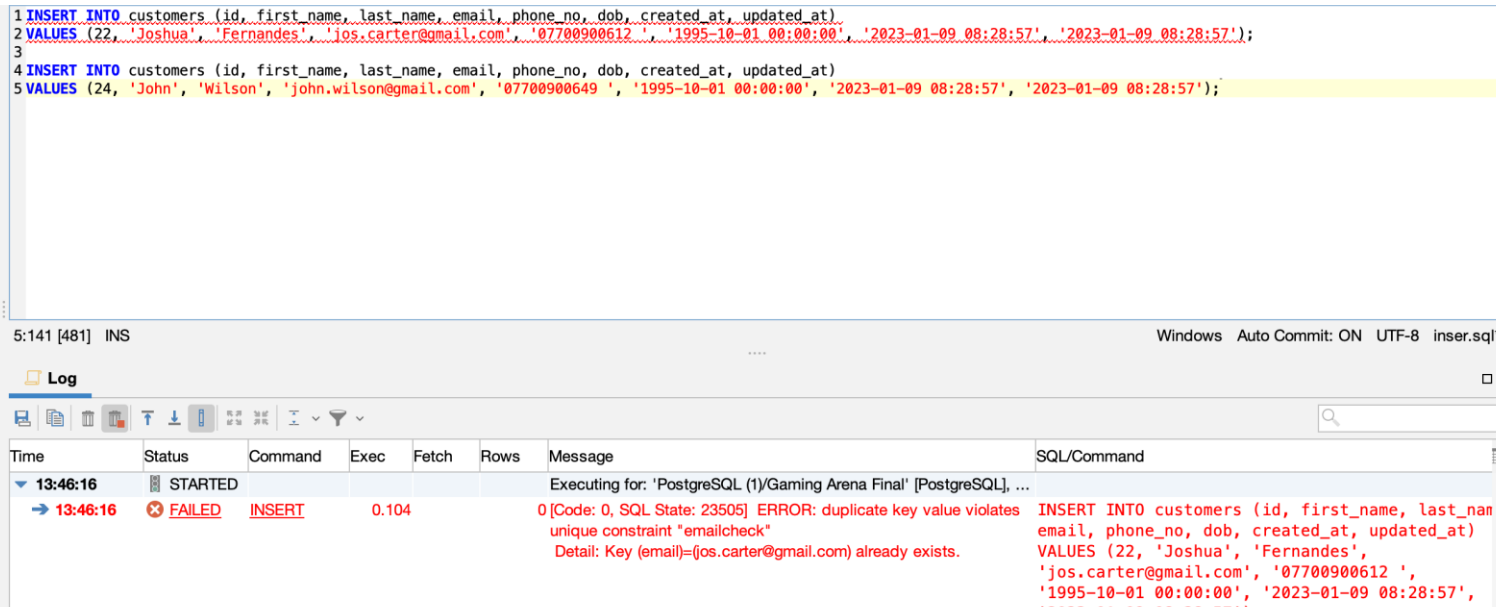


```
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at, updated_at)
VALUES (22, 'Joshua', 'Fernandes', 'joshua.fernandes@gmail.com', '07700900888 ', '1995-10-01 00:00:00', '2023-01-09 08:28:57', '2023-01-09 08:28:57');

INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at, updated_at)
VALUES (23, 'Jos', 'Carter', 'jos.carter@gmail.com', '07700900889 ', '1995-10-01 00:00:00', '2023-01-09 08:28:57', '2023-01-09 08:28:57');
```

```
Time      Status   Command Exec  Fetch Rows Message                                                   SQL/Command
13:32:27  STARTED                                Executing for: 'PostgreSQL (1)/Gaming Arena Final' [PostgreSQL], ...
13:32:27  FAILED   INSERT  0.141            0 [Code: 0, SQL State: 23505]  ERROR: duplicate key value violates   INSERT INTO customers (id, first_name, last_name, email,
                                              unique constraint "phonenumbercheck"                                 phone_no, dob, created_at, updated_at)
                                               Detail: Key (phone_no)=(07700900888 ) already exists.               VALUES (22, 'Joshua', 'Fernandes',
                                                                                                                   'joshua.fernandes@gmail.com', '07700900888 ', '1995-10-01
                                                                                                                   00:00:00', '2023-01-09 08:28:57', '2023-01-09 08:28:57')
13:32:27  SUCCESS  INSERT  0.020            1 OK                                                                   INSERT INTO customers (id, first_name, last_name, email, pho...
13:32:27  FINISHED         0.161  0         1 Success: 1  Failed: 1
```

This is a test to check that a phone number is unique as it can later be used for 2 factor authentication. Figure shows that if a phone number is the same a row will not be added.

6.3 Email Test

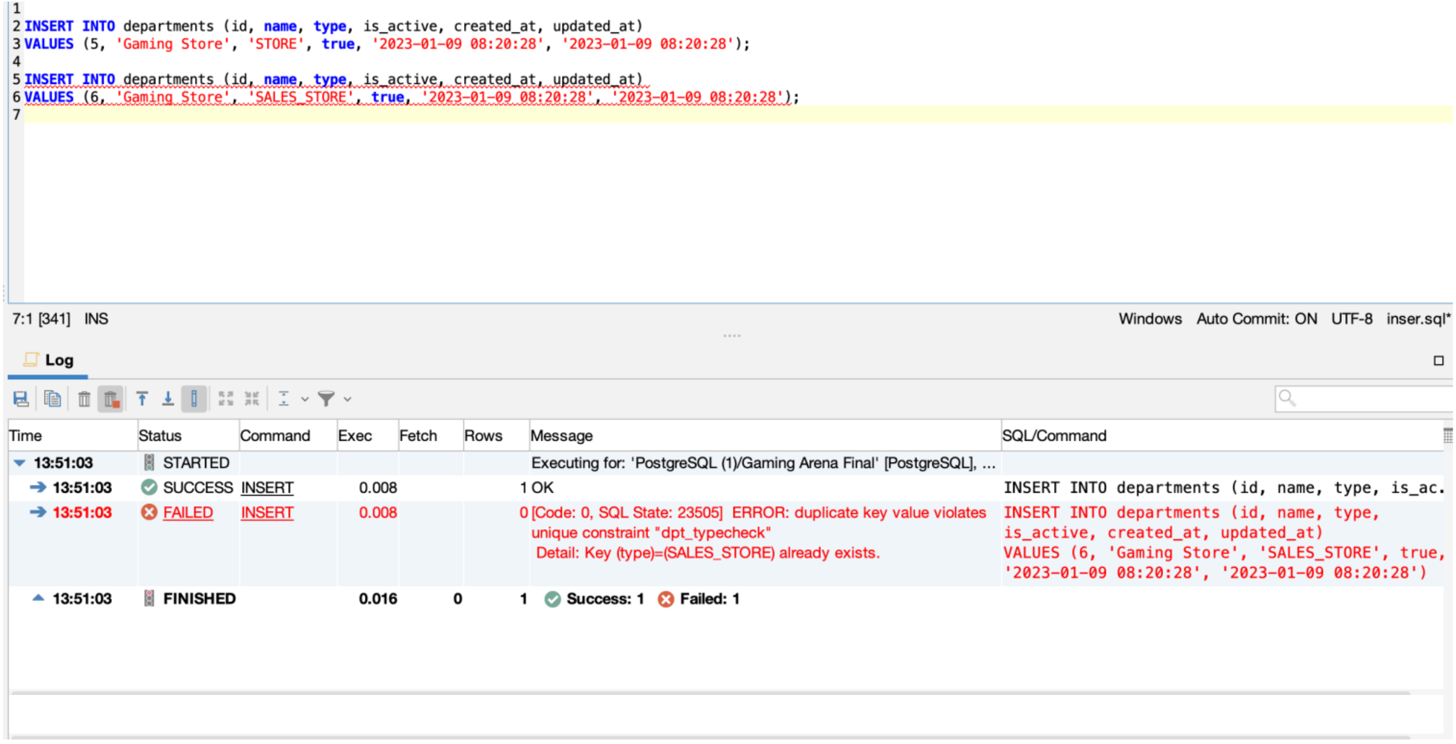


```
INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at, updated_at)
VALUES (22, 'Joshua', 'Fernandes', 'jos.carter@gmail.com', '07700900612 ', '1995-10-01 00:00:00', '2023-01-09 08:28:57', '2023-01-09 08:28:57');

INSERT INTO customers (id, first_name, last_name, email, phone_no, dob, created_at, updated_at)
VALUES (24, 'John', 'Wilson', 'john.wilson@gmail.com', '07700900649 ', '1995-10-01 00:00:00', '2023-01-09 08:28:57', '2023-01-09 08:28:57');
```

```
Time      Status   Command Exec  Fetch Rows Message                                                   SQL/Command
13:46:16  STARTED                                Executing for: 'PostgreSQL (1)/Gaming Arena Final' [PostgreSQL], ...
13:46:16  FAILED   INSERT  0.104            0 [Code: 0, SQL State: 23505]  ERROR: duplicate key value violates   INSERT INTO customers (id, first_name, last_nam
                                              unique constraint "emailcheck"                                       email, phone_no, dob, created_at, updated_at)
                                               Detail: Key (email)=(jos.carter@gmail.com) already exists.          VALUES (22, 'Joshua', 'Fernandes',
                                                                                                                   'jos.carter@gmail.com', '07700900612 ',
                                                                                                                   '1995-10-01 00:00:00', '2023-01-09 08:28:57',
```

This test shows how if the email is the same as an email from a different row then the row will not be added. Figure shows that if the email is unique then the row will be added successfully.

6.4Department Type Test



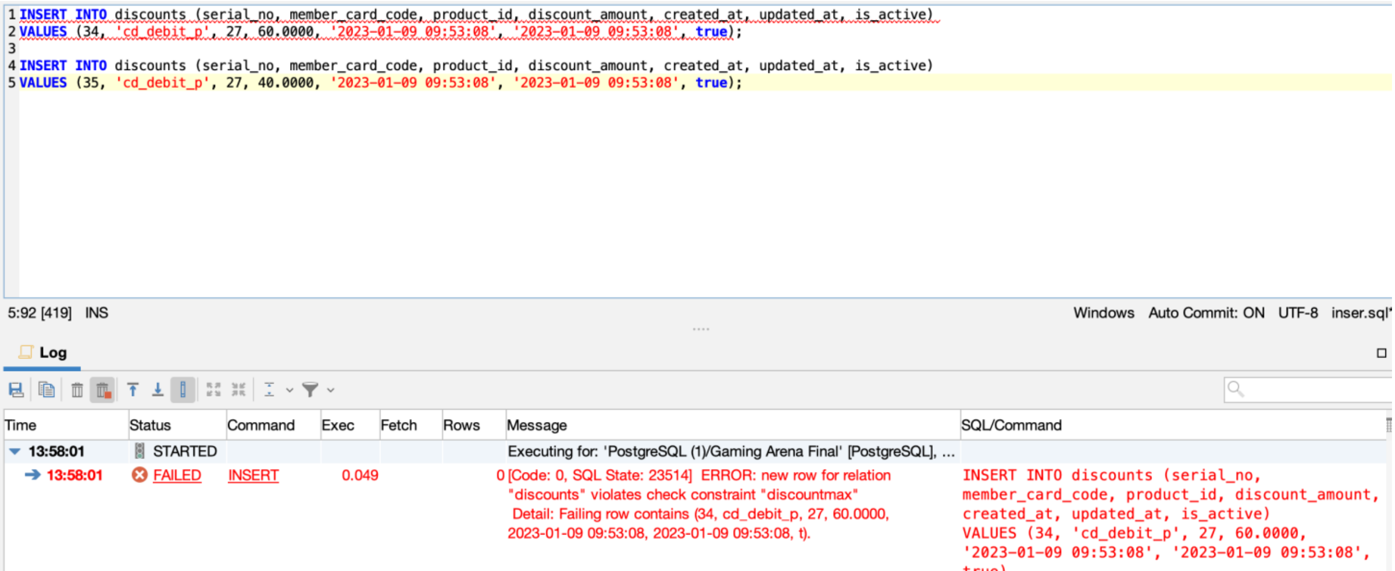
```
INSERT INTO departments (id, name, type, is_active, created_at, updated_at)
VALUES (5, 'Gaming Store', 'STORE', true, '2023-01-09 08:20:28', '2023-01-09 08:20:28');

INSERT INTO departments (id, name, type, is_active, created_at, updated_at)
VALUES (6, 'Gaming Store', 'SALES_STORE', true, '2023-01-09 08:20:28', '2023-01-09 08:20:28');
```

```
7:1 [341]  INS                                                    Windows  Auto Commit: ON  UTF-8  inser.sql*
```

```
Time        Status     Command  Exec   Fetch  Rows  Message                                                   SQL/Command
▼ 13:51:03  STARTED                                 Executing for: 'PostgreSQL (1)/Gaming Arena Final' [PostgreSQL], ...
 → 13:51:03 ✓ SUCCESS  INSERT   0.008               1 OK                                                      INSERT INTO departments (id, name, type, is_ac.
 → 13:51:03 ✗ FAILED   INSERT   0.008               0 [Code: 0, SQL State: 23505]  ERROR: duplicate key value violates   INSERT INTO departments (id, name, type,
                                                      unique constraint "dpt_typecheck"                       is_active, created_at, updated_at)
                                                       Detail: Key (type)=(SALES_STORE) already exists.       VALUES (6, 'Gaming Store', 'SALES_STORE', true,
                                                                                                              '2023-01-09 08:20:28', '2023-01-09 08:20:28')
 ▲ 13:51:03  FINISHED           0.016  0       1     ✓ Success: 1  ✗ Failed: 1
```

This figure shows that of a department type is the same as one that already exists a row will not be created and shows that if it does not exist a row will be added.

6.5Discounts Test



```
INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount, created_at, updated_at, is_active)
VALUES (34, 'cd_debit_p', 27, 60.0000, '2023-01-09 09:53:08', '2023-01-09 09:53:08', true);

INSERT INTO discounts (serial_no, member_card_code, product_id, discount_amount, created_at, updated_at, is_active)
VALUES (35, 'cd_debit_p', 27, 40.0000, '2023-01-09 09:53:08', '2023-01-09 09:53:08', true);
```

```
5:92 [419]  INS                                                   Windows  Auto Commit: ON  UTF-8  inser.sql*
```

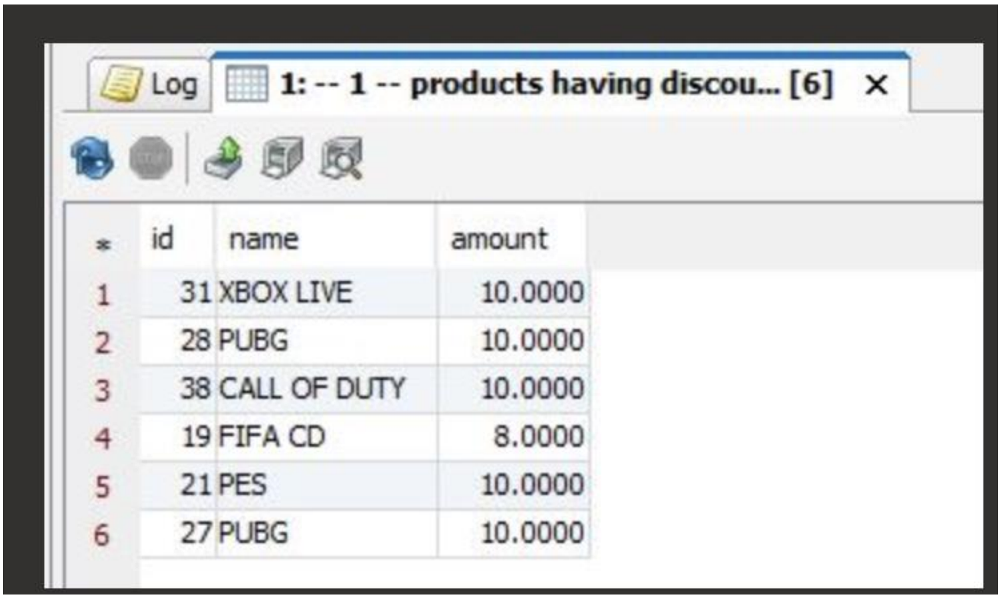
```
Time        Status     Command  Exec   Fetch  Rows  Message                                                   SQL/Command
▼ 13:58:01  STARTED                                 Executing for: 'PostgreSQL (1)/Gaming Arena Final' [PostgreSQL], ...
 → 13:58:01 ✗ FAILED   INSERT   0.049               0 [Code: 0, SQL State: 23514]  ERROR: new row for relation   INSERT INTO discounts (serial_no,
                                                       "discounts" violates check constraint "discountmax"    member_card_code, product_id, discount_amount,
                                                       Detail: Failing row contains (34, cd_debit_p, 27, 60.0000,   created_at, updated_at, is_active)
                                                      2023-01-09 09:53:08, 2023-01-09 09:53:08, t).           VALUES (34, 'cd_debit_p', 27, 60.0000,
                                                                                                              '2023-01-09 09:53:08', '2023-01-09 09:53:08',
                                                                                                              true)
```

One of the business rules is that a discount amount cannot be higher than 50. This check will test that there are no amounts higher than 50. As shown in this figure if the discount amount is higher than 50 a row will not be added.

7.Queries

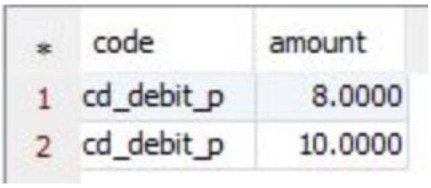
Queries by Muhammad Salman

```
-- 1
-- products having discount > 5
select p.id, p.name, d.discount_amount as "amount"  from products p
join discounts d on d.product_id = p.id
where d.discount_amount > 5
group by 1,2,3;
```



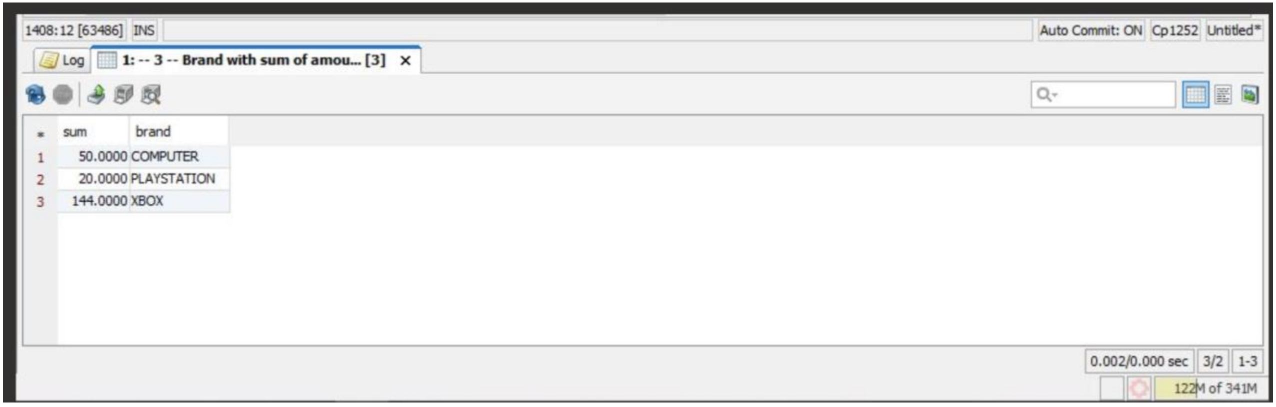
*	id	name	amount
1	31	XBOX LIVE	10.0000
2	28	PUBG	10.0000
3	38	CALL OF DUTY	10.0000
4	19	FIFA CD	8.0000
5	21	PES	10.0000
6	27	PUBG	10.0000

```
-- 2
--To fetch the list card_code and discount amount having discount > 5
select bc.code, d.discount_amount as "amount"  from products p
join discounts d on d.product_id = p.id
join b_cards bc on bc.code = d.member_card_code
where d.discount_amount >  5
group by 1,2;
```



*	code	amount
1	cd_debit_p	8.0000
2	cd_debit_p	10.0000

```
-- 3
--  Brand with sum of amount  with sattus completed
select SUM(t.amount), p.brand from transactions t
join orders o on o.id = t.order_id and o.status = 'COMPLETED'
join order_details od on od.order_id = o.id
join products p on p.id = od.product_id
group by 2;
```

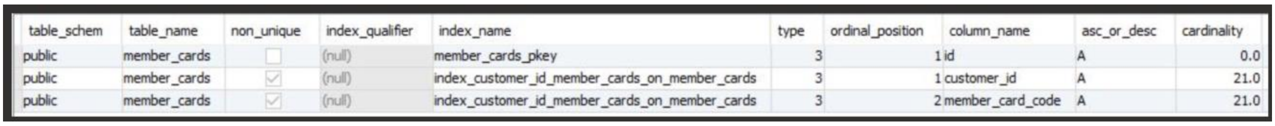


*	sum	brand
1	50.0000	COMPUTER
2	20.0000	PLAYSTATION
3	144.0000	XBOX

Index Query by Muhammad Salman

```
create index concurrently "index_customer_id_member_cards_on_member_cards"
```

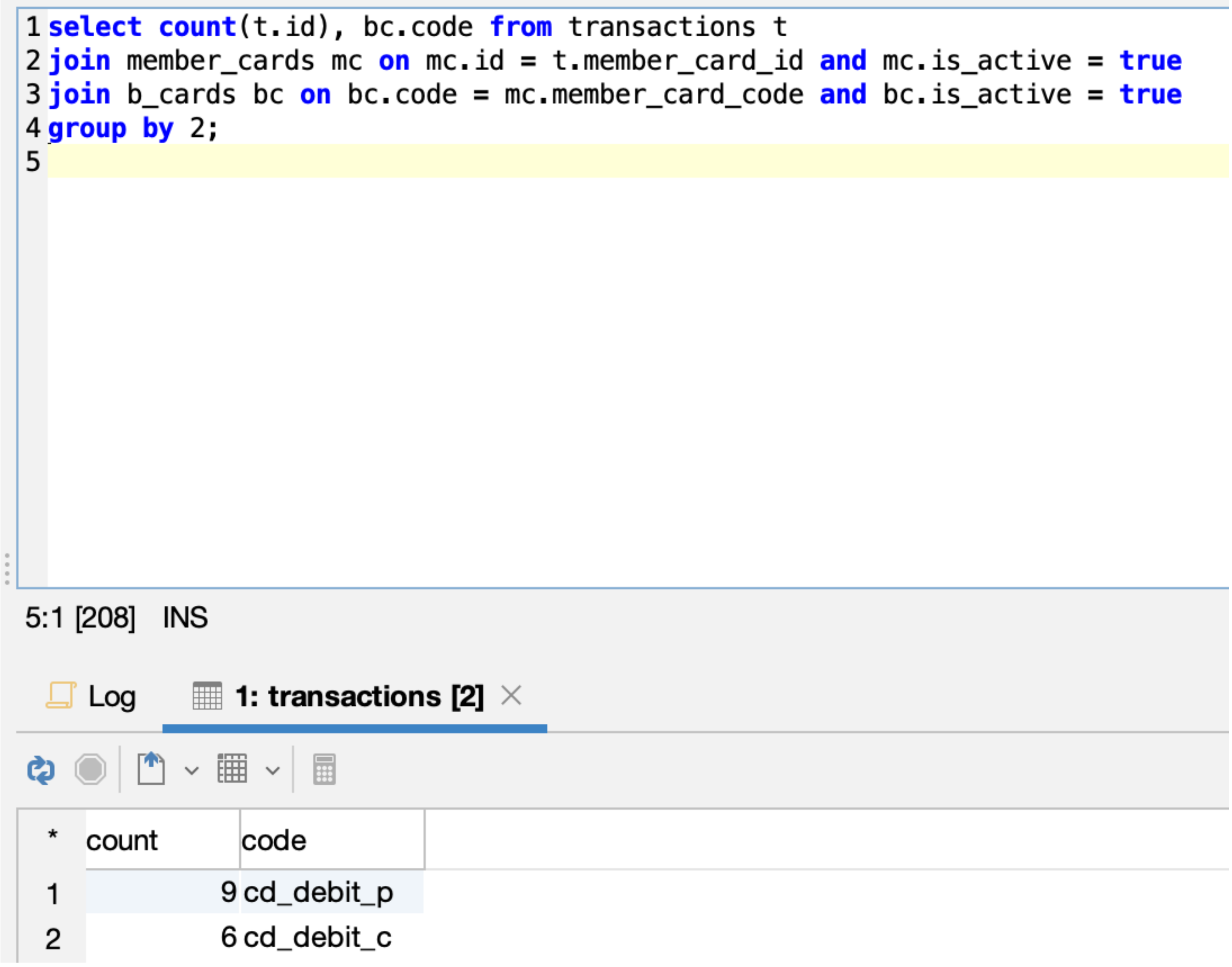
```
on member_cards using btree (customer_id, member_card_code);
--query
select count(t.id), bc.code from transactions t
join member_cards mc on mc.id = t.member_card_id and mc.is_active = true
join b_cards bc on bc.code = mc.member_card_code and bc.is_active = true
group by 2;
```



table_schem	table_name	non_unique	index_qualifier	index_name	type	ordinal_position	column_name	asc_or_desc	cardinality
public	member_cards	□	(null)	member_cards_pkey	3	1	id	A	0.0
public	member_cards	☑	(null)	index_customer_id_member_cards_on_member_cards	3	1	customer_id	A	21.0
public	member_cards	☑	(null)	index_customer_id_member_cards_on_member_cards	3	2	member_card_code	A	21.0

Queries by Joshua Fernandes

```
-- 4
-- Total number of transactions using different card types
select count(t.id), bc.code from transactions t
join member_cards mc on mc.id = t.member_card_id and mc.is_active = true
join b_cards bc on bc.code = mc.member_card_code and bc.is_active = true
group by 2;
```



```
select count(t.id), bc.code from transactions t
join member_cards mc on mc.id = t.member_card_id and mc.is_active = true
join b_cards bc on bc.code = mc.member_card_code and bc.is_active = true
group by 2;
```

*	count	code
1	9	cd_debit_p
2	6	cd_debit_c

-- 5

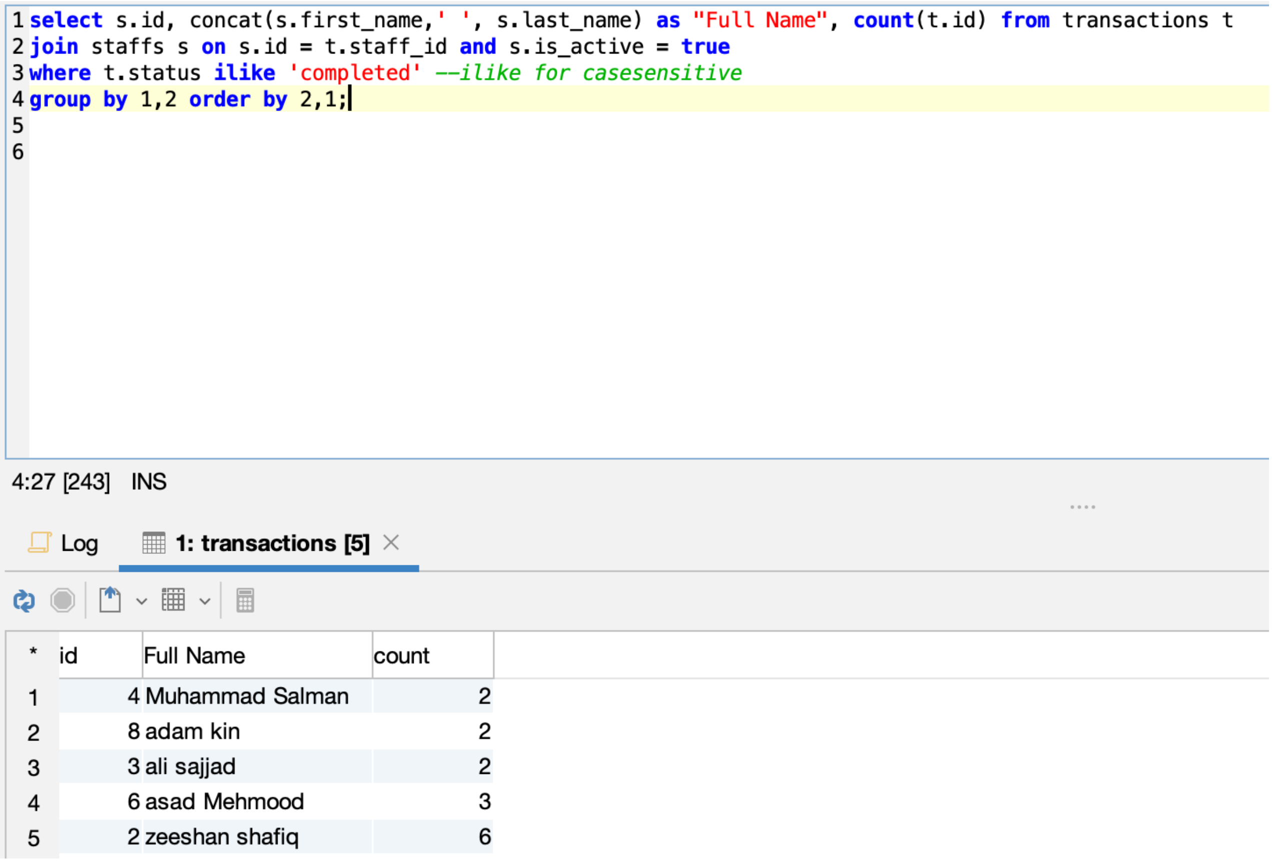
-- Total Number of transactions processed by staffs full name when status is complete

select s.id, concat(s.first_name,' ', s.last_name) as 'Full Name', count(t.id) from transactions t

join staffs s on s.id = t.staff_id and s.is_active = true

where t.status ilike 'completed' --ilike for casesensitive

group by 1,2 order by 2,1;



```
select s.id, concat(s.first_name,' ', s.last_name) as "Full Name", count(t.id) from transactions t
join staffs s on s.id = t.staff_id and s.is_active = true
where t.status ilike 'completed' --ilike for casesensitive
group by 1,2 order by 2,1;
```

*	id	Full Name	count
1	4	Muhammad Salman	2
2	8	adam kin	2
3	3	ali sajjad	2
4	6	asad Mehmood	3
5	2	zeeshan shafiq	6

-- 6

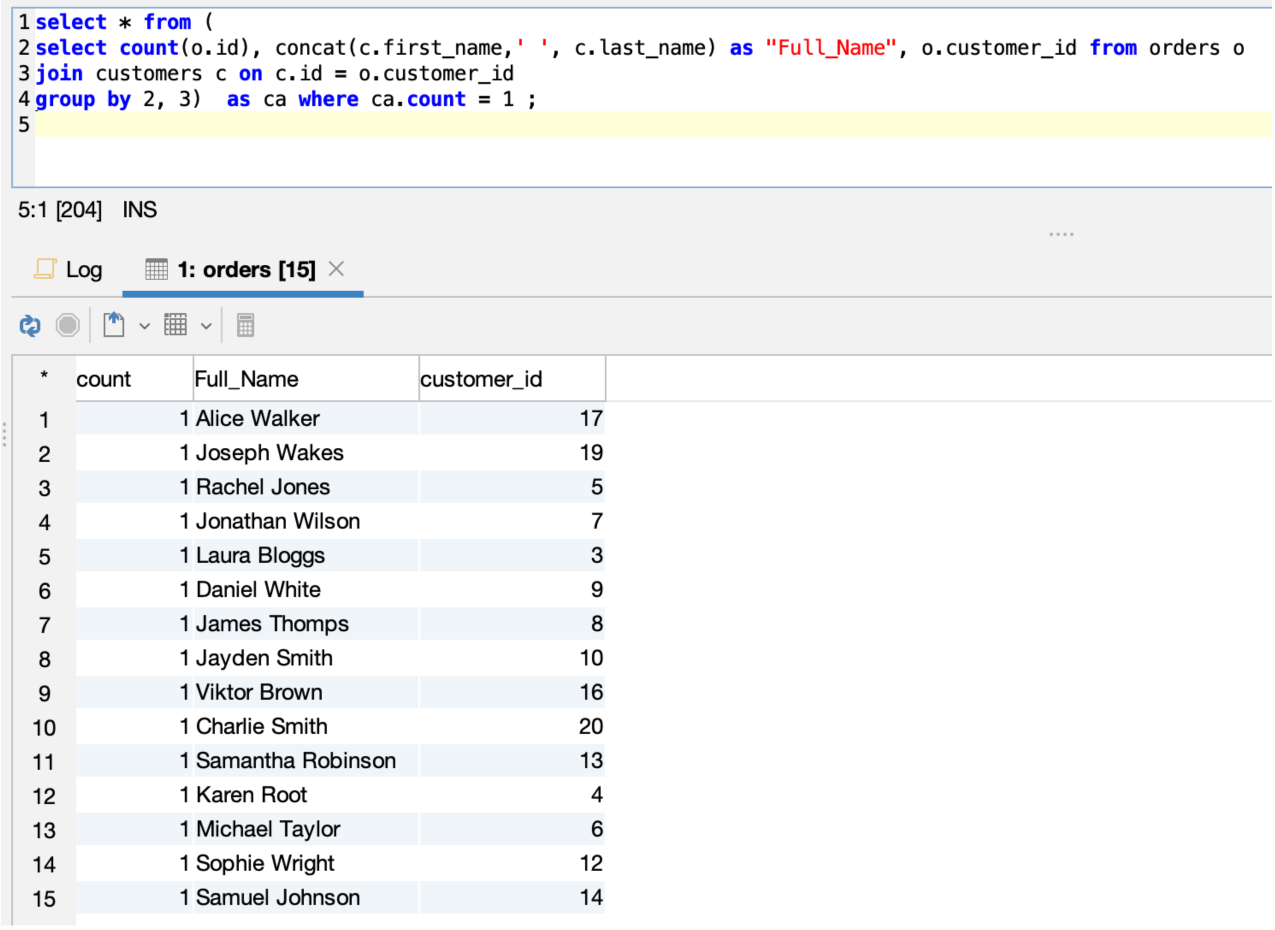
-- In order to get the loyal/repeat customer that have more orders we can search for the amount

select * from (

select count(o.id), concat(c.first_name,' ', c.last_name) as "Full_Name", o.customer_id from orders o

join customers c on c.id = o.customer_id

group by 2, 3) as ca where ca.count = 1 ;

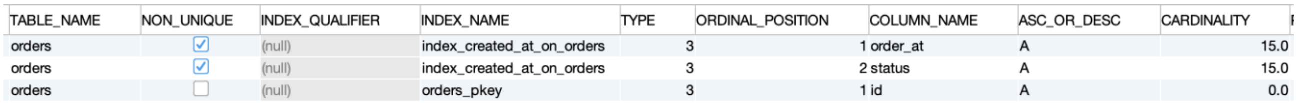


```
select * from (
select count(o.id), concat(c.first_name,' ', c.last_name) as "Full_Name", o.customer_id from orders o
join customers c on c.id = o.customer_id
group by 2, 3)  as ca where ca.count = 1 ;

```

*	count	Full_Name	customer_id
1	1	Alice Walker	17
2	1	Joseph Wakes	19
3	1	Rachel Jones	5
4	1	Jonathan Wilson	7
5	1	Laura Bloggs	3
6	1	Daniel White	9
7	1	James Thomps	8
8	1	Jayden Smith	10
9	1	Viktor Brown	16
10	1	Charlie Smith	20
11	1	Samantha Robinson	13
12	1	Karen Root	4
13	1	Michael Taylor	6
14	1	Sophie Wright	12
15	1	Samuel Johnson	14

Index Query by Joshua Fernandes

create index concurrently "index_created_at_on_orders"
on orders using btree (order_at, status);
select SUM(t.amount), p.brand from transactions t
join orders o on o.id = t.order_id and o.status = 'COMPLETED'
join order_details od on od.order_id = o.id
join products p on p.id = od.product_id
group by 2;



TABLE_NAME	NON_UNIQUE	INDEX_QUALIFIER	INDEX_NAME	TYPE	ORDINAL_POSITION	COLUMN_NAME	ASC_OR_DESC	CARDINALITY
orders	☑	(null)	index_created_at_on_orders	3	1	order_at	A	15.0
orders	☑	(null)	index_created_at_on_orders	3	2	status	A	15.0
orders	□	(null)	orders_pkey	3	1	id	A	0.0

Conclusion

This shows a database that has been created for gaming arena and a gaing store put into one. The level of complexity is quite high as there were two sides to this store. Going forward we could take a closer look at addresses so that we could focus on game delivery as well as looking at community events going on in the Arena.Databases are very effective to make sure the store work well together and by itself and helps keep track of anything going on in the store.It also makes it easier to have all the data in one place and in the future will help with marketing and promotions for games going forward.