University of Dhaka
Department of Computer Science and Engineering
CSE-2211: Database Management Systems-1
Lab Project

Lab Group: B (EVEN)

Ibna Afra Roza (Roll 16)
Mehedi Hasan (Roll 22)

Submitted To:
Associate Prof. Mr. Abu Ahmed Ferdaus, Dept. of CSE, University of Dhaka
Lecturer Redwan Ahmed Rizvee, Dept. of CSE, University of Dhaka

Submission Date: 17/7/25

Contents

1 Introduction

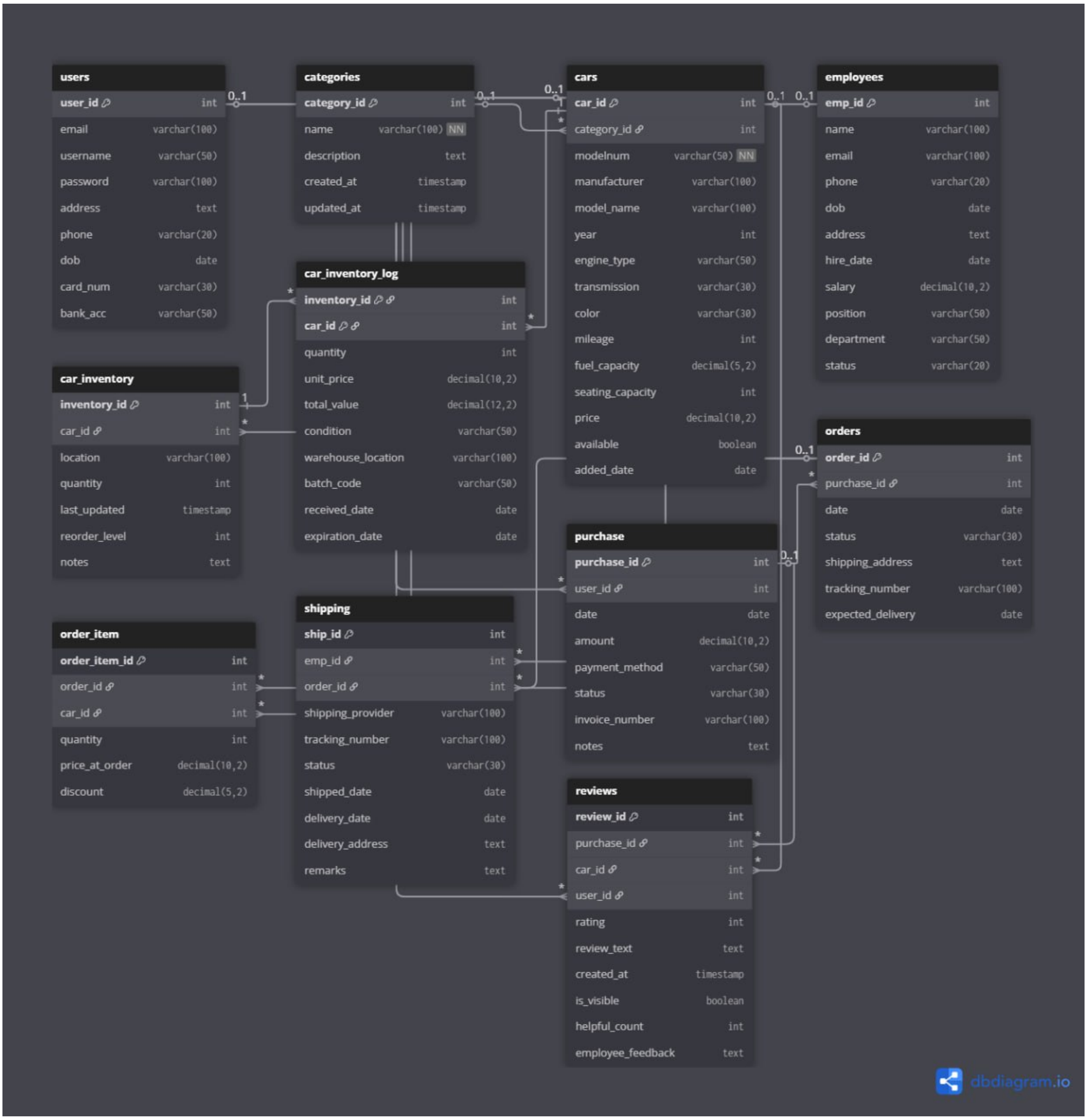
This project implements a comprehensive database system for a car dealership management platform. The system supports the management of users, employees, cars, inventory, purchases, orders, and reviews. It ensures data integrity with various constraints and supports complex queries to retrieve business insights. The backend is implemented in a relational DBMS (e.g., Oracle or PostgreSQL).

2 Database Schema Overview

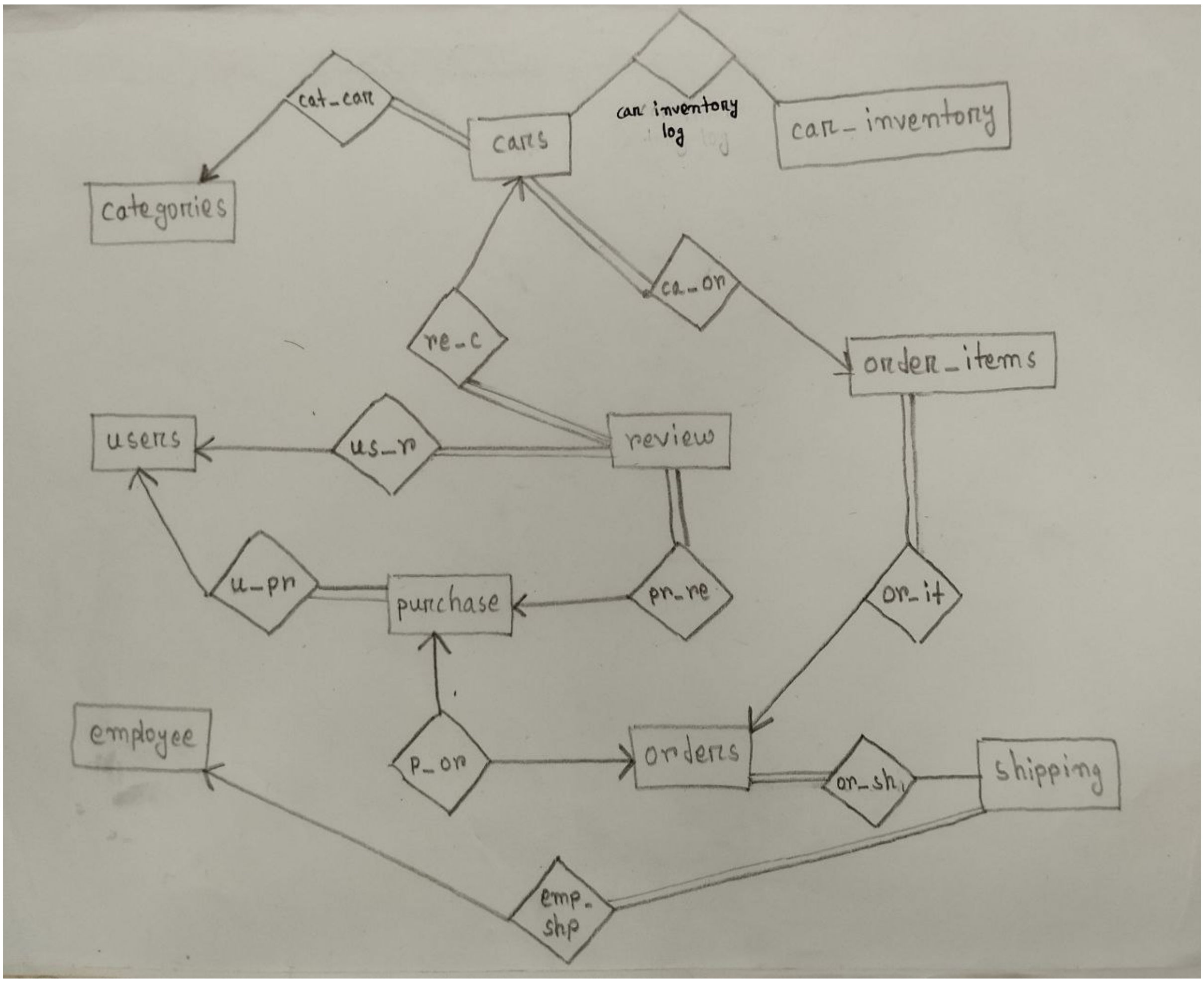
The system is composed of the following tables:

- **users**
- **categories**
- **cars**
- **employees**
- **car_inventory**
- **car_inventory_log**
- **purchase**
- **orders**
- **order_item**
- **shipping**
- **reviews**

3 Schema Diagram



4 ER Diagram


cat_car
cars
car inventory log
car_inventory
categories
ca_or
re_c
order_items
users
us_r
review
u_pr
purchase
pr_re
or_it
employee
p_or
orders
or_sh
shipping
emp_shp

5 SQL DDL Commands

5.1 users Table

```sql
CREATE TABLE users (
    user_id SERIAL PRIMARY KEY,
    email VARCHAR(100) UNIQUE,
    username VARCHAR(50) UNIQUE,
    password VARCHAR(100),
    address TEXT,
    phone VARCHAR(20),
    dob DATE,
    card_num VARCHAR(30),
    bank_acc VARCHAR(50)
);
```

5.2 categories Table

```sql
CREATE TABLE categories (
    category_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    created_at TIMESTAMP DEFAULT current_timestamp,
    updated_at TIMESTAMP DEFAULT current_timestamp
);
```

5.3 cars Table

```sql
CREATE TABLE cars (
    car_id SERIAL PRIMARY KEY,
    category_id INT REFERENCES categories(category_id),
    modelnum VARCHAR(50) NOT NULL,
    manufacturer VARCHAR(100),
    model_name VARCHAR(100),
    year INT,
    engine_type VARCHAR(50),
    transmission VARCHAR(30),
    color VARCHAR(30),
    mileage INT,
    fuel_capacity NUMERIC(5,2),
    seating_capacity INT,
    price NUMERIC(10,2),
    available BOOLEAN DEFAULT TRUE,
    added_date DATE DEFAULT CURRENT_DATE
);
```

5.4 employees Table

```sql
CREATE TABLE employees (
    emp_id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(20),
    dob DATE,
    address TEXT,
    hire_date DATE,
    salary NUMERIC(10,2),
    position VARCHAR(50),
    department VARCHAR(50),
    status VARCHAR(20) DEFAULT 'active'
);
```

5.5 car_inventory Table

```sql
CREATE TABLE car_inventory (
    inventory_id SERIAL PRIMARY KEY,
    car_id INT REFERENCES cars(car_id),
    location VARCHAR(100),
    quantity INT CHECK (quantity >= 0),
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    reorder_level INT DEFAULT 5,
    notes TEXT
);
```

5.6 car_inventory_log Table

```sql
CREATE TABLE car_inventory_log (
    inventory_id INT REFERENCES car_inventory(inventory_id),
    car_id INT REFERENCES cars(car_id),
    quantity INT CHECK (quantity >= 0),
    unit_price NUMERIC(10,2),
    total_value NUMERIC(12,2),
    condition VARCHAR(50),
    warehouse_location VARCHAR(100),
    batch_code VARCHAR(50),
    received_date DATE,
    expiration_date DATE,
    PRIMARY KEY (inventory_id, car_id)
);
```

5.7 purchase Table

```
CREATE TABLE purchase (
    purchase_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(user_id),
    date DATE DEFAULT CURRENT_DATE,
    amount NUMERIC(10, 2),
    payment_method VARCHAR(50),
    status VARCHAR(30) DEFAULT 'pending',
    invoice_number VARCHAR(100) UNIQUE,
    notes TEXT
);
```

5.8 orders Table

```
CREATE TABLE orders (
    order_id SERIAL PRIMARY KEY,
    purchase_id INT REFERENCES purchase(purchase_id),
    date DATE DEFAULT CURRENT_DATE,
    status VARCHAR(30) DEFAULT 'processing',
    shipping_address TEXT,
    tracking_number VARCHAR(100),
    expected_delivery DATE
);
```

5.9 order_item Table

```
CREATE TABLE order_item (
    order_item_id SERIAL PRIMARY KEY,
    order_id INT REFERENCES orders(order_id),
    car_id INT REFERENCES cars(car_id),
    quantity INT CHECK (quantity > 0),
    price_at_order NUMERIC(10, 2),
    discount NUMERIC(5,2) DEFAULT 0.00
);
```

5.10 shipping Table

```
CREATE TABLE shipping (
    ship_id SERIAL PRIMARY KEY,
    emp_id INT REFERENCES employees(emp_id),
    order_id INT REFERENCES orders(order_id),
    shipping_provider VARCHAR(100),
    tracking_number VARCHAR(100),
```

```sql
    status VARCHAR(30) DEFAULT 'pending',
    shipped_date DATE,
    delivery_date DATE,
    delivery_address TEXT,
    remarks TEXT
);
```

5.11 reviews Table

```sql
CREATE TABLE reviews (
    review_id SERIAL PRIMARY KEY,
    purchase_id INT REFERENCES purchase(purchase_id),
    car_id INT REFERENCES cars(car_id),
    user_id INT REFERENCES users(user_id),
    rating INT CHECK (rating >= 1 AND rating <= 5),
    review_text TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    is_visible BOOLEAN DEFAULT TRUE,
    helpful_count INT DEFAULT 0,
    employee_feedback TEXT
);
```

6 Data Samples

This section provides snapshots of the populated tables in our database to demonstrate the system's functionality and show actual data samples.

6.1 users table snapshot

user_id...	email varcha...	username...	password...	address...	phone v...	dob date	card_num...	bank_ac...	purchases	reviews
1	roza@gmail.c...	roza.afra	hashed_pass...	123 Dhaka...	880171234...	1990-05-15	1234-5678-9...	9876543210	purchases	reviev
2	mehedi@gmail...	hasanmehedi	hashed_pass...	456 Chitt...	880181234...	1985-11-22	9876-5432-1...	1234567890	purchases	reviev
3	sara@example...	sara.ahmed	hashed_pass...	789 Sylhe...	880191234...	1992-03-30	1122-3344-5...	5678901234	purchases	reviev
4	karim@exampl...	karim.uddin	hashed_pass...	101 Mirpu...	880161234...	1987-08-10	2233-4455-6...	6789012345	purchases	reviev
5	nabila@examp...	nabila.khan	hashed_pass...	202 Gulsh...	880172345...	1995-12-01	3344-5566-7...	7890123456	purchases	reviev
6	arif@example...	arif.hossain	hashed_pass...	303 Banan...	880182345...	1990-06-25	4455-6677-8...	8901234567	purchases	reviev
7	sumaiya@exam...	sumaiya.islam	hashed_pass...	404 Uttar...	880192345...	1993-04-15	5566-7788-9...	9012345678	purchases	reviev
8	rafsan@examp...	rafsan.jani	hashed_pass...	505 Dhanm...	880173456...	1989-09-20	6677-8899-0...	0123456789	purchases	reviev
9	ayesha@examp...	ayesha.begum	hashed_pass...	606 Motij...	880183456...	1991-11-11	7788-9900-1...	1234567891	purchases	reviev
10	tariq@exampl...	tariq.rahman	hashed_pass...	707 Barid...	880193456...	1986-07-07	8899-0011-2...	2345678901	purchases	reviev
11	zara@example...	zara.khan	hashed_pass...	808 Khilg...	880174567...	1994-02-28	9900-1122-3...	3456789012	purchases	reviev
12	imran@exampl...	imran.haque	hashed_pass...	909 Rampu...	880184567...	1988-10-10	0011-2233-4...	4567890123	purchases	reviev

Figure 1: Users Table Sample Data

6.2 categories table snapshot

category_id serial	name varcha...	description text	created_at timestamp	updated_at timestamp	cars
1	Sedan	Four-door passenger ca...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
2	SUV	Sport Utility Vehicles...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
3	Coupe	Two-door passenger car...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
4	Electric	Vehicles powered by el...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
5	Hatchback	Compact cars with a re...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
6	Convertible	Cars with retractable ...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
7	Minivan	Family-friendly vehicl...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
8	Truck	Heavy-duty vehicles fo...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
9	Luxury	Premium vehicles with ...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
10	Hybrid	Vehicles combining pet...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
11	Sports	High-performance cars ...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars
12	Van	Spacious vehicles for ...	2025-07-16 17:54:...	2025-07-16 17:54:14.27...	cars

Figure 2: Categories Table Sample Data

6.3 cars table snapshot

car_id s...	categor...	modelnum v...	manufacturer...	model_name v...	year integer	engine_type...	transmi...	color va...	mileage...
9	9	S-Class	Mercedes-Benz	S-Class	2023	Petrol	Automatic	Black	600
5	5	Golf GTI	Volkswagen	Golf	2022	Petrol	Manual	Black	1000
3	2	CR-V EX-L	Honda	CR-V	2024	Petrol	Automatic	Blue	150
11	11	Mustang	Ford	Mustang	2022	Petrol	Manual	Blue	1200
8	8	F-150	Ford	F-150	2023	Petrol	Automatic	Green	400
7	7	Odyssey	Honda	Odyssey	2024	Petrol	Automatic	Grey	200
2	3	911 Carrera S	Porsche	911 Carrera S	2022	Petrol	Automatic	Red	2000
12	12	Sprinter	Mercedes-Benz	Sprinter	2023	Diesel	Automatic	Silver	500
4	4	Leaf	Nissan	Leaf	2023	Electric	Automatic	Silver	300
10	10	Prius	Toyota	Prius	2024	Hybrid	Automatic	White	100
1	1	Model S	Tesla	Model S	2023	Electric	Automatic	White	500
6	6	MX-5 Miata	Mazda	MX-5	2023	Petrol	Manual	Yellow	800

Figure 3: Cars Table Sample Data

6.4 employees table snapshot

emp_id...	name varcha...	email varcha...	phone v...	dob date	address...	hire_date d...	salary nu...	position varch...	department varchar
1	Jamil Khan	jamil@exampl...	880151122...	1988-07-20	321 Banan...	2020-01-10	50000.00	Sales Manager	Sales
2	Fahim Ahmed	fahim@exampl...	880165566...	1995-02-14	654 Uttar...	2021-03-22	35000.00	Mechanic	Maintenance
3	Nusrat Jahan	nusrat@examp...	880177788...	1991-09-05	987 Khilg...	2019-05-01	45000.00	Logistics Coor...	Logistics
4	Asif Rahman	asif@example...	880152233...	1987-03-12	111 Dhanm...	2018-07-15	60000.00	Store Manager	Operations
5	Shabnam Begum	shabnam@exam...	880166677...	1990-01-30	222 Gulsh...	2022-02-10	40000.00	Sales Associate	Sales
6	Rashed Uddin	rashed@examp...	880178899...	1985-05-25	333 Mirpu...	2020-11-01	38000.00	Technician	Maintenance
7	Laila Akter	laila@exampl...	880153344...	1993-08-18	444 Rampu...	2021-09-05	42000.00	HR Specialist	Human Resources
8	Kamal Hossain	kamal@exampl...	880167788...	1989-12-20	555 Motij...	2019-03-15	47000.00	Accountant	Finance
9	Sadia Islam	sadia@exampl...	880179900...	1992-06-10	666 Barid...	2020-06-20	45000.00	Marketing Mana...	Marketing
10	Tareq Mahmud	tareq@exampl...	880154455...	1986-04-05	777 Sylhe...	2021-01-12	39000.00	Delivery Driver	Logistics
11	Mousumi Rahman	mousumi@exam...	880168899...	1994-10-15	888 Chitt...	2022-04-01	36000.00	Customer Servi...	Support
12	Anwar Hossain	anwar@exampl...	880170011...	1988-11-11	999 Dhaka...	2019-08-10	41000.00	Inventory Clerk	Operations

Figure 4: Employees Table Sample Data

6.5 car_inventory table snapshot

inventory_id...	car_id i...	location varchar(100)	quantity...	last_updated timestamp	reorder_level...	notes text	car
1	1	Showroom A	2	2025-07-16 17:54:53.34...	5	New arrivals	car
2	2	Warehouse B	1	2025-07-16 17:54:53.34...	5	Luxury sports car	car
3	3	Showroom B	3	2025-07-16 17:54:53.34...	5	Popular family SUV	car
4	4	Showroom A	4	2025-07-16 17:54:53.34...	5	Eco-friendly electric	car
5	5	Warehouse C	2	2025-07-16 17:54:53.34...	5	Sporty hatchback	car
6	6	Showroom B	1	2025-07-16 17:54:53.34...	5	Convertible fun	car
7	7	Showroom A	3	2025-07-16 17:54:53.34...	5	Family minivan	car
8	8	Warehouse B	2	2025-07-16 17:54:53.34...	5	Heavy-duty truck	car
9	9	Showroom C	1	2025-07-16 17:54:53.34...	5	Luxury sedan	car
10	10	Showroom A	4	2025-07-16 17:54:53.34...	5	Hybrid efficiency	car
11	11	Warehouse C	2	2025-07-16 17:54:53.34...	5	Classic sports car	car
12	12	Showroom B	2	2025-07-16 17:54:53.34...	5	Spacious van	car

Figure 5: Car Inventory Table Sample Data

6.6 car_inventory_log table snapshot

inventory_id...	car_id i...	quantity...	unit_price nu...	total_value nu...	conditio...	warehouse_locati...	batch_code...	received_date date
1	1	2	8000000.00	16000000.00	new	Showroom A	NULL	2025-07-10
2	2	1	11500000.00	11500000.00	new	Warehouse B	NULL	2025-07-05
3	3	3	4000000.00	12000000.00	new	Showroom B	NULL	2025-07-12
4	4	4	3200000.00	12800000.00	new	Showroom A	NULL	2025-07-08
5	5	2	3000000.00	6000000.00	new	Warehouse C	NULL	2025-07-06
6	6	1	3800000.00	3800000.00	new	Showroom B	NULL	2025-07-09
7	7	3	4800000.00	14400000.00	new	Showroom A	NULL	2025-07-11
8	8	2	5800000.00	11600000.00	new	Warehouse B	NULL	2025-07-07
9	9	1	14500000.00	14500000.00	new	Showroom C	NULL	2025-07-10
10	10	4	3600000.00	14400000.00	new	Showroom A	NULL	2025-07-08
11	11	2	5200000.00	10400000.00	new	Warehouse C	NULL	2025-07-06
12	12	2	6800000.00	13600000.00	new	Showroom B	NULL	2025-07-09

Figure 6: Car Inventory Log Table Sample Data

6.7 purchase table snapshot

purchase_id...	user_id...	date date	amount num...	payment_method...	status va...	invoice_number varchar...	notes text
1	1	2025-07-16	12000000.00	Credit Card	paid	INV-20250713-001	NULL
2	2	2025-07-16	4500000.00	Bank Transfer	paid	INV-20250713-002	NULL
3	3	2025-07-16	3500000.00	Credit Card	paid	INV-20250713-003	NULL
4	4	2025-07-16	4000000.00	Bank Transfer	paid	INV-20250713-004	NULL
5	5	2025-07-16	15000000.00	Credit Card	paid	INV-20250713-005	NULL
6	6	2025-07-16	3800000.00	Bank Transfer	paid	INV-20250713-006	NULL
7	7	2025-07-16	5500000.00	Credit Card	paid	INV-20250713-007	NULL
8	8	2025-07-16	7000000.00	Bank Transfer	paid	INV-20250713-008	NULL
9	9	2025-07-16	3200000.00	Credit Card	paid	INV-20250713-009	NULL
10	10	2025-07-16	8500000.00	Bank Transfer	paid	INV-20250713-010	NULL
11	11	2025-07-16	5000000.00	Credit Card	paid	INV-20250713-011	NULL
12	12	2025-07-16	6000000.00	Bank Transfer	paid	INV-20250713-012	NULL

Figure 7: Purchase Table Sample Data

6.8 orders table snapshot

order_i...	purchase_id...	date date	status va...	shipping_address text	tracking_num...	expected_delivery date	order_items
1	1	2025-07-16	shipped	123 Dhaka St, Dhaka	TRK-987654321	2025-07-20	order_items
2	2	2025-07-16	processing	456 Chittagong Rd, Chi...	TRK-123456789	2025-07-25	order_items
3	3	2025-07-16	shipped	789 Sylhet Rd, Sylhet	TRK-456789123	2025-07-22	order_items
4	4	2025-07-16	processing	101 Mirpur Rd, Dhaka	TRK-789123456	2025-07-26	order_items
5	5	2025-07-16	shipped	202 Gulshan Ave, Dhaka	TRK-321654987	2025-07-21	order_items
6	6	2025-07-16	processing	303 Banani Rd, Dhaka	TRK-654987321	2025-07-27	order_items
7	7	2025-07-16	shipped	404 Uttara St, Dhaka	TRK-987321654	2025-07-23	order_items
8	8	2025-07-16	processing	505 Dhanmondi Rd, Dhaka	TRK-147258369	2025-07-28	order_items
9	9	2025-07-16	shipped	606 Motijheel St, Dhaka	TRK-258369147	2025-07-24	order_items
10	10	2025-07-16	processing	707 Baridhara Rd, Dhaka	TRK-369147258	2025-07-29	order_items
11	11	2025-07-16	shipped	808 Khilgaon St, Dhaka	TRK-741852963	2025-07-25	order_items
12	12	2025-07-16	processing	909 Rampura Rd, Dhaka	TRK-852963741	2025-07-30	order_items

Figure 8: Orders Table Sample Data

6.9 order_item table snapshot

order_item_id se...	order_i...	car_id i...	quantity integer	price_at_order numeric(1...	discount numeric(5, 2)	order	car
1	1	2	1	12000000.00	0.00	order	car
2	2	3	1	4500000.00	0.00	order	car
3	3	4	1	3500000.00	0.00	order	car
4	4	6	1	4000000.00	0.00	order	car
5	5	9	1	15000000.00	0.00	order	car
6	6	10	1	3800000.00	0.00	order	car
7	7	11	1	5500000.00	0.00	order	car
8	8	12	1	7000000.00	0.00	order	car
9	9	5	1	3200000.00	0.00	order	car
10	10	1	1	8500000.00	0.00	order	car
11	11	7	1	5000000.00	0.00	order	car
12	12	8	1	6000000.00	0.00	order	car

Figure 9: Order Item Table Sample Data

6.10 shipping table snapshot

ship_id s...	emp_id...	order_i...	shippin...	tracking_num...	status va...	shipped_date...	delivery_...	delivery_address text	remark
1	3	1	DHL	TRK-987654321	shipped	2025-07-13	2025-07-20	123 Dhaka St, Dhaka	NULL
2	10	3	FedEx	TRK-456789123	shipped	2025-07-14	2025-07-22	789 Sylhet Rd, Sylhet	NULL
3	3	5	UPS	TRK-321654987	shipped	2025-07-13	2025-07-21	202 Gulshan Ave, Dhaka	NULL
4	10	7	DHL	TRK-987321654	shipped	2025-07-15	2025-07-23	404 Uttara St, Dhaka	NULL
5	3	9	FedEx	TRK-258369147	shipped	2025-07-14	2025-07-24	606 Motijheel St, Dhaka	NULL
6	10	11	UPS	TRK-741852963	shipped	2025-07-15	2025-07-25	808 Khilgaon St, Dhaka	NULL
7	3	2	DHL	TRK-123456789	processing	2025-07-16	NULL	456 Chittagong Rd, Chi...	NULL
8	10	4	FedEx	TRK-789123456	processing	2025-07-16	NULL	101 Mirpur Rd, Dhaka	NULL
9	3	6	UPS	TRK-654987321	processing	2025-07-16	NULL	303 Banani Rd, Dhaka	NULL
10	10	8	DHL	TRK-147258369	processing	2025-07-16	NULL	505 Dhanmondi Rd, Dhaka	NULL
11	3	10	FedEx	TRK-369147258	processing	2025-07-16	NULL	707 Baridhara Rd, Dhaka	NULL
12	10	12	UPS	TRK-852963741	processing	2025-07-16	NULL	909 Rampura Rd, Dhaka	NULL

Figure 10: Shipping Table Sample Data

6.11 reviews table snapshot

	review_id...	purchase_id...	car_id i...	user_id...	rating i...	review_text text	created_at timestamp	is_visible boolean	helpful_c
	1	1	2	1	5	Absolutely love t…	2025-07-16 17:55:…	TRUE	0
	2	2	3	2	4	Great car for the…	2025-07-16 17:55:…	TRUE	0
	3	3	4	3	5	The Nissan Leaf i…	2025-07-16 17:55:…	TRUE	0
	4	4	6	4	4	The Mazda MX-5 is…	2025-07-16 17:55:…	TRUE	0
	5	5	9	5	5	The Mercedes S-Cl…	2025-07-16 17:55:…	TRUE	0
	6	6	10	6	4	Toyota Prius is g…	2025-07-16 17:55:…	TRUE	0
	7	7	11	7	5	The Mustang has a…	2025-07-16 17:55:…	TRUE	0
	8	8	12	8	4	The Sprinter is s…	2025-07-16 17:55:…	TRUE	0
	9	9	5	9	4	The Golf GTI is s…	2025-07-16 17:55:…	TRUE	0
	10	10	1	10	5	Tesla Model S is …	2025-07-16 17:55:…	TRUE	0
	11	11	7	11	4	Honda Odyssey is …	2025-07-16 17:55:…	TRUE	0
	12	12	8	12	5	Ford F-150 is tou…	2025-07-16 17:55:…	TRUE	0
	13	1	2	1	5	Absolutely love t…	2025-07-16 17:55:…	TRUE	0
	14	2	3	2	4	Great car for the…	2025-07-16 17:55:…	TRUE	0
	15	3	4	3	5	The Nissan Leaf i…	2025-07-16 17:55:…	TRUE	0
	16	4	6	4	4	The Mazda MX-5 is…	2025-07-16 17:55:…	TRUE	0
	17	5	9	5	5	The Mercedes S-Cl…	2025-07-16 17:55:…	TRUE	0

Figure 11: Reviews Table Sample Data

7 Query Examples with Relational Algebra

7.1 Available Cars with Category (NATURAL JOIN)

SQL:

```sql
SELECT c.model_name, c.manufacturer, c.year, c.price,
    cat.name AS category_name
FROM cars c
NATURAL JOIN categories cat
WHERE c.available = TRUE;
```

Relational Algebra:

$$\pi_{\text{model\_name, manufacturer, year, price, name}} (\sigma_{\text{available = TRUE}}(\text{CARS} \bowtie \text{CATEGORIES})) \tag{1}$$

model_name	manufacturer	year	price	category_name
Model S	Tesla	2023	8500000.00	Sedan
CR-V	Honda	2024	4500000.00	SUV
911 Carrera S	Porsche	2022	12000000.00	Coupe
Leaf	Nissan	2023	3500000.00	Electric
Golf	Volkswagen	2022	3200000.00	Hatchback
MX-5	Mazda	2023	4000000.00	Convertible
Odyssey	Honda	2024	5000000.00	Minivan
F-150	Ford	2023	6000000.00	Truck
S-Class	Mercedes-Benz	2023	15000000.00	Luxury
Prius	Toyota	2024	3800000.00	Hybrid
Mustang	Ford	2022	5500000.00	Sports
Sprinter	Mercedes-Benz	2023	7000000.00	Van

Figure 12: Query Output: Available Cars with Category

7.2 All Users and Their Purchases (LEFT OUTER JOIN)

SQL:

```sql
SELECT u.username, p.purchase_id, p.amount, p.date
FROM users u
LEFT OUTER JOIN purchase p ON u.user_id = p.user_id
WHERE p.status = 'paid' OR p.purchase_id IS NULL;
```

Relational Algebra:

$$\text{Paid\_Purchases} \leftarrow \sigma_{\text{status='paid'} \vee \text{purchase\_id IS NULL}}(\text{PURCHASE})$$
$$\text{Result} \leftarrow \pi_{\text{username, purchase\_id, amount, date}}(\text{USERS LEFT JOIN}_{\text{user\_id}}\text{Paid\_Purchases})$$

username	purchase_id	amount	date
jane_smith			
2			
alice_wilson			
john_doe			

Figure 13: Query Output: All Users and Their Purchases

7.3 Order Details with Car Information (USING Clause)

SQL:

```sql
SELECT
  o.order_id,
```

```
  o.date       AS order_date,
  c.model_name
FROM orders o
JOIN order_item oi USING (order_id)
JOIN cars       c USING (car_id)
WHERE o.status = 'processing';
```

Relational Algebra:

$$\begin{aligned}
\text{Proc_Orders} &\leftarrow \sigma_{\text{status='processing'}}(\text{ORDERS}) \\
\text{Order_Items} &\leftarrow \text{Proc_Orders} \bowtie_{\text{order_id}} \text{ORDER_ITEM} \\
\text{Result} &\leftarrow \pi_{\text{order_id, date, model_name}}\big(\text{Order_Items} \bowtie_{\text{car_id}} \text{CARS}\big)
\end{aligned}$$

order_id	order_date	model_name
2	2025-07-16	CR-V
4	2025-07-16	MX-5
6	2025-07-16	Prius
8	2025-07-16	Sprinter
10	2025-07-16	Model S
12	2025-07-16	F-150
2	2025-07-16	911 Carrera S

Figure 14: Query Output: Order Details in Processing Status

7.4 Users with Completed Purchases (EXISTS)

SQL:

```
SELECT u.username, u.email
FROM users u
WHERE EXISTS (
   SELECT 1
   FROM purchase p
   WHERE p.user_id = u.user_id
   AND p.status = 'paid'
);
```

Relational Algebra:

$$\pi_{\text{username, email}} \left(\text{USERS} \bowtie_{\text{user\_id}} \sigma_{\text{status='paid'}}(\text{PURCHASE})\right) \tag{2}$$

username	email
imran.haque	imran@example.com
sara.ahmed	sara@example.com
zara.khan	zara@example.com
rafsan.jani	rafsan@example.com
tariq.rahman	tariq@example.com
ayesha.begum	ayesha@example.com
sumaiya.islam	sumaiya@example.com
roza.afra	roza@gmail.com
nabila.khan	nabila@example.com
karim.uddin	karim@example.com
hasanmehedi	mehedi@gmail.com
arif.hossain	arif@example.com

Figure 15: Query Output: Users with Completed Purchases

7.5 Cars More Expensive Than All Cars in a Category (ALL)

SQL:

```sql
SELECT model_name , price
FROM cars
WHERE price > ALL (
    SELECT price
    FROM cars
    WHERE category_id = 1
)
ORDER BY price DESC;
```

Relational Algebra:

$$\text{Cat1\_Prices} \leftarrow \pi_{\text{price}}(\sigma_{\text{category\_id}=1}(\text{CARS}))$$
$$\text{Result} \leftarrow \pi_{\text{model\_name, price}}(\text{CARS}) - \pi_{\text{model\_name, price}}(\sigma_{\text{CARS.price} \leq \text{p1.price}}(\text{CARS} \times \rho_{\text{p1}}(\text{Cat1\_Prices})))$$

model_name	price
S-Class	15000000.00
911 Carrera S	12000000.00

Figure 16: Query Output: Cars More Expensive Than All Cars in a Category

7.6 Employees and Number of Orders Handled (Scalar Subquery)

SQL:

```sql
SELECT
  e.emp_id,
  e.name,
  e.position,
  (
    SELECT COUNT(*)
    FROM shipping s
    WHERE s.emp_id = e.emp_id
  ) AS total_shipments,
  (
    SELECT COUNT(*)
    FROM shipping s
    WHERE s.emp_id = e.emp_id
      AND s.status = 'delivered'
  ) AS deliveries_completed
FROM employees e
WHERE e.status = 'active'
ORDER BY deliveries_completed DESC, total_shipments DESC,
   e.name;
```

Relational Algebra:

$$\begin{aligned}
\text{ActiveEmp} &\leftarrow \sigma_{\text{status='active'}}(\text{EMPLOYEES})\\
\text{ShipCnt} &\leftarrow \gamma_{\text{emp_id; COUNT(*)}\rightarrow\text{total_shipments}}(\text{SHIPPING})\\
\text{DelCnt} &\leftarrow \gamma_{\text{emp_id; COUNT(*)}\rightarrow\text{deliveries_completed}}\big(\sigma_{\text{status='delivered'}}(\text{SHIPPING})\big)\\
\text{Temp} &\leftarrow \text{ActiveEmp} \bowtie_{\text{emp_id}} \text{ShipCnt} \bowtie_{\text{emp_id}} \text{DelCnt}\\
\text{Result} &\leftarrow \pi_{\text{emp_id, name, position, total_shipments, deliveries_completed}}(\text{Temp})
\end{aligned}$$

emp_id	name	position	total_shipments	deliveries_completed
3	Nusrat Jahan	Logistics Coordinator	6	0
10	Tareq Mahmud	Delivery Driver	6	0
12	Anwar Hossain	Inventory Clerk	0	0
4	Asif Rahman	Store Manager	0	0
16	Bob Brown	Sales Associate	0	0
2	Fahim Ahmed	Mechanic	0	0
1	Jamil Khan	Sales Manager	0	0
14	Jane Smith	Sales Associate	0	0
13	John Doe	Manager	0	0
8	Kamal Hossain	Accountant	0	0
7	Laila Akter	HR Specialist	0	0
11	Mousumi Rahman	Customer Service	0	0
6	Rashed Uddin	Technician	0	0

Figure 17: Query Output: Employee Shipment Counts

7.7 Top 5 Most Reviewed Cars (WITH/CTE)

SQL:

```sql
WITH CarReviews AS (
    SELECT c.model_name, c.manufacturer, COUNT(r.review_id)
       AS review_count
    FROM cars c
    LEFT JOIN reviews r ON c.car_id = r.car_id
    GROUP BY c.model_name, c.manufacturer
)
SELECT model_name, manufacturer, review_count
FROM CarReviews
ORDER BY review_count DESC
LIMIT 5;
```

Relational Algebra:

$$\text{CarReviews} \leftarrow \gamma_{\text{model\_name, manufacturer;COUNT(review\_id)}\rightarrow\text{review\_count}}$$
$$(\text{CARS LEFT JOIN}_{\text{car\_id}}\text{REVIEWS})$$

model_name	manufacturer	review_count
Model S	Tesla	2
Leaf	Nissan	2
MX-5	Mazda	2
Golf	Volkswagen	2
F-150	Ford	2

Figure 18: Query Output: Top 5 Most Reviewed Cars

7.8 Available Cars and Inventory Quantities (INNER JOIN)

SQL:

```sql
SELECT c.model_name, c.manufacturer, ci.quantity, ci.location
FROM cars c
INNER JOIN car_inventory ci ON c.car_id = ci.car_id
WHERE c.available = TRUE AND ci.quantity > 0
ORDER BY ci.quantity DESC;
```

Relational Algebra:

$$\begin{aligned}&\tau_{\text{quantity DESC}}\big(\\&\quad\pi_{\text{model\_name, manufacturer, quantity, location}}\big(\\&\quad\quad\sigma_{\text{available=TRUE}\wedge\text{quantity}>0}\big(\\&\quad\quad\quad\text{CARS}\bowtie_{\text{car\_id}}\text{CAR\_INVENTORY}\big)\big)\big)\end{aligned}\tag{3}$$

model_name	manufacturer	quantity	location
Prius	Toyota	4	Showroom A
Leaf	Nissan	4	Showroom A
Odyssey	Honda	3	Showroom A
CR-V	Honda	3	Showroom B
Sprinter	Mercedes-Benz	2	Showroom B
Golf	Volkswagen	2	Warehouse C
F-150	Ford	2	Warehouse B
Mustang	Ford	2	Warehouse C
Model S	Tesla	2	Showroom A
MX-5	Mazda	1	Showroom B
911 Carrera S	Porsche	1	Warehouse B
S-Class	Mercedes-Benz	1	Showroom C

Figure 19: Query Output: Available Cars and Inventory Quantities

7.9 Employees and Their Shipping Records (RIGHT OUTER JOIN)

SQL:

```sql
SELECT
  e.emp_id,
  e.name               AS employee_name,
  e.department,
  s.ship_id,
  s.shipping_provider,
  s.status             AS shipping_status,
  s.shipped_date,
  s.delivery_date
FROM shipping s
RIGHT OUTER JOIN employees e ON s.emp_id = e.emp_id
WHERE e.status = 'active'
ORDER BY s.shipped_date DESC NULLS LAST;
```

Relational Algebra:

$$
\begin{aligned}
\text{ActiveEmp} &\leftarrow \sigma_{\text{status='active'}}(\text{EMPLOYEES}) \\
\text{EmpShip} &\leftarrow \text{ActiveEmp} \;\;\rtimes_{\text{emp\_id=emp\_id}}\;\; \text{SHIPPING} \\
\text{Result} &\leftarrow \pi_{\substack{\text{emp\_id, name, department,} \\ \text{ship\_id, shipping\_provider,} \\ \text{shipping\_status, shipped\_date, delivery\_date}}}(\text{EmpShip})
\end{aligned}
$$

emp_id	employee_name	department	ship_id	shipping_provider	shipping_status	shipped_date	delivery_d
10	Tareq Mahmud	Logistics	12	UPS	processing	2025-07-16	
10	Tareq Mahmud	Logistics	8	FedEx	processing	2025-07-16	
3	Nusrat Jahan	Logistics	9	UPS	processing	2025-07-16	
10	Tareq Mahmud	Logistics	10	DHL	processing	2025-07-16	
3	Nusrat Jahan	Logistics	11	FedEx	processing	2025-07-16	
3	Nusrat Jahan	Logistics	7	DHL	processing	2025-07-16	
10	Tareq Mahmud	Logistics	4	DHL	shipped	2025-07-15	2025-07-2
10	Tareq Mahmud	Logistics	6	UPS	shipped	2025-07-15	2025-07-2
3	Nusrat Jahan	Logistics	5	FedEx	shipped	2025-07-14	2025-07-2
10	Tareq Mahmud	Logistics	2	FedEx	shipped	2025-07-14	2025-07-2
3	Nusrat Jahan	Logistics	1	DHL	shipped	2025-07-13	2025-07-2
3	Nusrat Jahan	Logistics	3	UPS	shipped	2025-07-13	2025-[illegible]

Figure 20: Query Output: Employees with Shipping Records

7.10 Visible Reviews with User and Car Details (Multiple JOIN)

SQL:

```sql
SELECT r.review_id, r.rating, r.review_text, u.username,
    c.model_name
FROM reviews r
JOIN users u USING (user_id)
JOIN cars c USING (car_id)
WHERE r.is_visible = TRUE
ORDER BY r.created_at DESC;
```

Relational Algebra:

$$\begin{aligned}&\tau_{\text{created\_at DESC}}\big(\\ &\quad \pi_{\text{review\_id, rating, review\_text, username, model\_name}}\big(\\ &\quad\quad \sigma_{\text{is\_visible=TRUE}}\big(\\ &\quad\quad\quad (\text{REVIEWS} \bowtie_{\text{user\_id}} \text{USERS}) \bowtie_{\text{car\_id}} \text{CARS})))\end{aligned} \tag{4}$$

review_id	rating	review_text	username	model_name
24	5	Ford F-150 is tough and reliable. Great for heavy-duty tasks.	imran.haque	F-150
13	5	Absolutely love the Porsche! The red color is stunning and the performance is incredible.	roza.afra	911 Carrera S
14	4	Great car for the family. The blue color looks sleek and the handling is smooth.	hasanmehedi	CR-V
15	5	The Nissan Leaf is perfect for eco-conscious drivers. Super quiet and efficient.	sara.ahmed	Leaf
16	4	The Mazda MX-5 is so fun to drive, especially with the top down!	karim.uddin	MX-5
17	5	The Mercedes S-Class is pure luxury. Worth every penny.	nabila.khan	S-Class
18	4	Toyota Prius is great on fuel. Perfect for city driving.	arif.hossain	Prius
19	5	The Mustang has amazing power and style. Love the blue color.	sumaiya.islam	Mustang
20	4	The Sprinter is spacious and reliable for large groups.	rafsan.jani	Sprinter
21	4	The Golf GTI is sporty and practical. Great for daily use.	ayesha.begum	Golf
22	5	Tesla Model S is a game-changer. The tech is mind-blowing.	tariq.rahman	Model S
23	4	Honda Odyssey is perfect for family trips. Very comfortable.	zara.khan	Odyssey

Figure 21: Query Output: Visible Reviews with User and Car Details

7.11 Electric or Hybrid Cars (Pattern Matching)

SQL:

```sql
SELECT model_name, manufacturer, engine_type
FROM cars
WHERE engine_type ~* '^(electric|hybrid)$'
  AND available = TRUE
ORDER BY model_name;
```

Relational Algebra:

$$
\begin{aligned}
&\tau_{\text{model\_name}}(\\
&\quad \pi_{\text{model\_name, manufacturer, engine\_type}}(\\
&\quad\quad \sigma_{\text{available=TRUE}\wedge(\text{engine\_type='Electric'}\vee\text{engine\_type='Hybrid'})}(\\
&\quad\quad\quad \text{CARS})))
\end{aligned} \tag{5}
$$

model_name	manufacturer	engine_type
B-Class	Mercedes	hybrid
Corolla	Toyota	Hybrid
Fusion	Ford	hybrid
ID.4	Volkswagen	electric
Leaf	Nissan	electric
Leaf	Nissan	Electric
Model S	Tesla	Electric
Model S	Tesla	electric
Prius	Toyota	hybrid
Prius	Toyota	Hybrid
Volt	Chevrolet	electric

Figure 22: Query Output: Electric or Hybrid Cars

7.12 Insert a New Car (INSERT with Subquery)

SQL:

```sql
INSERT INTO cars (
 category_id, modelnum, manufacturer, model_name, year,
   engine_type, transmission, color, mileage, fuel_capacity,
      seating_capacity, price, available, added_date
     )
 VALUES (
     (SELECT category_id FROM categories WHERE name = '1'),
 '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12',
    TRUE,
CURRENT_DATE  )
RETURNING car_id, model_name, price;
```

car_id	model_name	price
13	4	12.00

Figure 23: Query Output: Insert a New Car

7.13 Register a New User (INSERT)

SQL:

```
INSERT INTO users (
  email, username, password, address, phone, dob, card_num,
  bank_acc
     )
        VALUES ('1', '2', '3', '4', '5', '2024-01-01', '7',
           '8')
           RETURNING user_id, username, email;
```

user_id	username	email
13	2	1

Figure 24: Query Output: Register a New User

7.14 Update Car Price and Availability (UPDATE)

SQL:

```
UPDATE cars
SET price = 1,
available = false,
added_date = CURRENT_DATE
WHERE car_id = 3
RETURNING car_id, model_name, price, available;
```

car_id	model_name	price	available
3	CR-V	1.00	f

Figure 25: Query Output: Update Car Price and Availability

7.15 Cars Cheaper Than Those in a Category (ANY Subquery)

SQL:

```
SELECT
  c.car_id,
  c.model_name,
  c.manufacturer,
  c.price
FROM cars c
```

```sql
WHERE c.price < ANY (
  SELECT c2.price
  FROM cars c2
  WHERE c2.category_id = category_id
)
ORDER BY c.price ASC;
```

Relational Algebra:

$$\begin{aligned}
\text{PriceSet} &\leftarrow \pi_{\text{price}}\big(\sigma_{\text{category_id}=:\text{category_id}}(\text{CARS})\big)\\
\text{Cheaper} &\leftarrow \sigma_{\exists p\in\text{PriceSet}\ (\text{CARS.price}<p)}(\text{CARS})\\
\text{Result} &\leftarrow \pi_{\text{car_id, model_name, manufacturer, price}}(\text{Cheaper})
\end{aligned}$$

car_id	model_name	manufacturer	price
3	CR-V	Honda	1.00
13	4	3	12.00
14	Corolla	Toyota	20000.00
15	Civic	Honda	22000.00
20	Civic	Honda	22999.99
30	Civic	Honda	22999.99
19	Prius	Toyota	24999.99
29	Prius	Toyota	24999.99
22	Fusion	Ford	28999.99
32	Fusion	Ford	28999.99
34	Leaf	Nissan	28999.99
24	Leaf	Nissan	28999.99
23	Volt	Chevrolet	34999.99

Figure 26: Query Output: Cars Cheaper Than Category Benchmark

7.16 Delete User by Email (DELETE)

SQL:

```sql
DELETE FROM users
WHERE email = 'john.doe@example.com'
RETURNING user_id, username, email;
```

user_id	username	email
14	john_doe	john.doe@example.com

Figure 27: Query Output: Delete User by Email

8 Views

Views are virtual tables that provide simplified access to complex data combinations. We have created two essential views for our car dealership platform.

8.1 Car Inventory Summary View

This view aggregates inventory statistics for each car, providing a comprehensive overview of their inventory performance.

```sql
CREATE VIEW car_inventory_summary AS
SELECT
    c.car_id,
    c.model_name,
    c.manufacturer,
    c.price,
    COALESCE(SUM(ci.quantity), 0) as total_quantity,
    COUNT(ci.inventory_id) as locations_count,
    AVG(ci.quantity) as avg_quantity_per_location
FROM cars c
LEFT JOIN car_inventory ci ON c.car_id = ci.car_id
GROUP BY c.car_id, c.model_name, c.manufacturer, c.price;
```

Usage Example: Find cars with low inventory levels across all locations to identify reorder needs.

```sql
SELECT model_name, manufacturer, total_quantity,
   locations_count
FROM car_inventory_summary
WHERE total_quantity < 10 OR locations_count < 2
ORDER BY total_quantity ASC;
```

model_name	manufacturer	total_quantity	locations_count
4	3	0	0
S-Class	Mercedes-Benz	1	1
911 Carrera S	Porsche	1	1
MX-5	Mazda	1	1
Golf	Volkswagen	2	1
Model S	Tesla	2	1
F-150	Ford	2	1
Sprinter	Mercedes-Benz	2	1
Mustang	Ford	2	1
CR-V	Honda	3	1
Odyssey	Honda	3	1
Leaf	Nissan	4	1
Prius	Toyota	4	1

Figure 28: Car Inventory Summary View Result: Low inventory cars

8.2 Customer Purchase History View

This view aggregates purchase statistics for each customer, providing insights into customer behavior and purchase patterns.

```sql
CREATE VIEW customer_purchase_history AS
SELECT
    u.user_id,
    u.username,
    u.email,
    COUNT(p.purchase_id) as total_purchases,
    COALESCE(SUM(p.amount), 0) as total_spent,
    AVG(p.amount) as avg_purchase_amount,
    MAX(p.date) as last_purchase_date,
    COUNT(CASE WHEN p.status = 'paid' THEN 1 END) as
       completed_purchases
FROM users u
LEFT JOIN purchase p ON u.user_id = p.user_id
GROUP BY u.user_id, u.username, u.email;
```

textbfUsage Example: List all customers and their total purchases, sorted by most recent purchase date.

```sql
SELECT username, email, total_purchases, total_spent,
   last_purchase_date
FROM customer_purchase_history
ORDER BY last_purchase_date DESC NULLS LAST;
```

username	email	total_purchases	total_spent	last_purchase_date
rafsan.jani	rafsan@example.com	1	7000000.00	2025-07-16
tariq.rahman	tariq@example.com	1	8500000.00	2025-07-16
arif.hossain	arif@example.com	1	3800000.00	2025-07-16
roza.afra	roza@gmail.com	2	12015000.00	2025-07-16
nabila.khan	nabila@example.com	1	15000000.00	2025-07-16
karim.uddin	karim@example.com	1	4000000.00	2025-07-16
hasanmehedi	mehedi@gmail.com	2	4520000.00	2025-07-16
zara.khan	zara@example.com	1	5000000.00	2025-07-16
ayesha.begum	ayesha@example.com	1	3200000.00	2025-07-16
sumaiya.islam	sumaiya@example.com	1	5500000.00	2025-07-16
imran.haque	imran@example.com	1	6000000.00	2025-07-16
sara.ahmed	sara@example.com	2	3500000.00	2025-07-16

Figure 29: Customer Purchase History View Result: All customers sorted by last purchase date

9 Functional Dependency and Normalization

9.1 Functional Dependencies

Below we list the principal functional dependencies for each table in our schema. Each table is treated in its own subsection.

9.1.1 users

$$\underline{\text{user\_id}} \rightarrow \{\text{email, username, password, address, phone, dob, card\_num, bank\_acc}\}$$

9.1.2 categories

$$\underline{\text{category\_id}} \rightarrow \{\text{name, description, created\_at, updated\_at}\}$$

9.1.3 cars

$$\underline{\text{car\_id}} \rightarrow \{\text{category\_id, modelnum, manufacturer, model\_name, year, engine\_type, transmission, color, mileage, fuel\_capacity, seating\_capacity, price, available, added\_date}\}$$

9.1.4 employees

emp_id → {name, email, phone, dob, address, hire_date, salary, position, department, status}

9.1.5 car_inventory

inventory_id → {car_id, location, quantity, last_updated, reorder_level, notes}

9.1.6 car_inventory_log

(inventory_id, car_id) → {quantity, unit_price, total_value, condition, warehouse_location, batch_code, received_date, expiration_date}

9.1.7 purchase

purchase_id → {user_id, date, amount, payment_method, status, invoice_number, notes}

9.1.8 orders

order_id → {purchase_id, date, status, shipping_address, tracking_number, expected_delivery}

9.1.9 order_item

order_item_id → {order_id, car_id, quantity, price_at_order, discount}

9.1.10 shipping

ship_id → {emp_id, order_id, shipping_provider, tracking_number, status, shipped_date, delivery_date, delivery_address, remarks}

9.1.11 reviews

review_id → {purchase_id, car_id, user_id, rating, review_text, created_at, is_visible, helpful_count, employee_feedback}

9.2 Normalization Process

Now, let's walk through the normalization steps for each of the tables.

9.2.1 1st Normal Form (1NF)

A table is in 1NF if:

- All columns contain atomic (indivisible) values.
- Each column contains values of a single type.
- Each column contains only one value per record.

In our schema, all tables already satisfy 1NF because:

- Each column has atomic values (e.g., no arrays or sets).
- Each attribute has a single value for each record (e.g., "address" does not contain multiple addresses).

9.2.2 2nd Normal Form (2NF)

A table is in 2NF if:

- It is in 1NF.
- Every non-prime attribute is fully functionally dependent on the primary key.

For example:

- In the **purchase** table, the attributes `user_id, date, amount, payment_method, status, invoice_number, notes` depend on the `purchase_id` (the primary key). All non-prime attributes are functionally dependent on the primary key.
- In the **orders** table, the attributes `purchase_id, date, status, shipping_address, tracking_number, expected_delivery` depend on `order_id`, which is the primary key.

All the tables comply with 2NF because there are no partial dependencies (i.e., no non-prime attributes depend on only part of a composite primary key).

9.2.3 3rd Normal Form (3NF)

A table is in 3NF if:

- It is in 2NF.
- No transitive dependency exists, i.e., non-prime attributes do not depend on other non-prime attributes.

In our schema:

- The **users** table has no transitive dependencies, as the primary key `user_id` determines all other attributes.
- The **cars** table has no transitive dependencies either. Attributes like `modelnum, manufacturer, year` depend solely on `car_id`.

Hence, all tables are in 3NF as there are no transitive dependencies.

9.3 BCNF Checking

A relation is in Boyce-Codd Normal Form (BCNF) if for every non-trivial functional dependency $X \to Y$, X is a superkey. This means that for every functional dependency, the determinant (the set on the left-hand side) must be a candidate key. We will now check whether each table is in BCNF by analyzing the functional dependencies.

9.3.1 1. `users` Table

- Functional Dependency:
  `user_id`→ `email, username, password, address, phone, dob, card_num, bank_acc`
- The primary key is `user_id`, which determines all other attributes. Since `user_id` is a superkey, this table is in BCNF.

9.3.2 2. `categories` Table

- Functional Dependency:
  `category_id`→ `name, description, created_at, updated_at`
- The primary key is `category_id`, which determines all other attributes. Since `category_id` is a superkey, this table is in BCNF.

9.3.3 3. `cars` Table

- Functional Dependency:
  `car_id`→ `category_id, modelnum, manufacturer, model_name, year, engine_type, transmission, color, mileage, fuel_capacity, seating_capacity, price, available, added_date`
- The primary key is `car_id`, which determines all other attributes. Since `car_id` is a superkey, this table is in BCNF.

9.3.4 4. `employees` Table

- Functional Dependency:
  `emp_id`→ `name, email, phone, dob, address, hire_date, salary, position, department, status`
- The primary key is `emp_id`, which determines all other attributes. Since `emp_id` is a superkey, this table is in BCNF.

9.3.5 5. `car_inventory` Table

- Functional Dependency:
  `inventory_id`→ `car_id, location, quantity, last_updated, reorder_level, notes`
- The primary key is `inventory_id`, which determines all other attributes. Since `inventory_id` is a superkey, this table is in BCNF.

9.3.6 6. `car_inventory_log` Table

- Functional Dependency:
  `(inventory_id, car_id)`→ `quantity, unit_price, total_value, condition, warehouse_location, batch_code, received_date, expiration_date`
- The primary key is `(inventory_id, car_id)`, which determines all other attributes. Since `(inventory_id, car_id)` is a superkey, this table is in BCNF.

9.3.7 7. `purchase` Table

- Functional Dependency:
  `purchase_id`→ `user_id, date, amount, payment_method, status, invoice_number, notes`

- The primary key is `purchase_id`, which determines all other attributes. Since `purchase_id` is a superkey, this table is in BCNF.

9.3.8 8. `orders` Table

- Functional Dependency:
`order_id` → `purchase_id, date, status, shipping_address, tracking_number, expected_delivery`
- The primary key is `order_id`, which determines all other attributes. Since `order_id` is a superkey, this table is in BCNF.

9.3.9 9. `order_item` Table

- Functional Dependency:
`order_item_id` → `order_id, car_id, quantity, price_at_order, discount`
- The primary key is `order_item_id`, which determines all other attributes. Since `order_item_id` is a superkey, this table is in BCNF.

9.3.10 10. `shipping` Table

- Functional Dependency:
`ship_id` → `emp_id, order_id, shipping_provider, tracking_number, status, shipped_date, delivery_date, delivery_address, remarks`
- The primary key is `ship_id`, which determines all other attributes. Since `ship_id` is a superkey, this table is in BCNF.

9.3.11 11. `reviews` Table

- Functional Dependency:
`review_id` → `purchase_id, car_id, user_id, rating, review_text, created_at, is_visible, helpful_count, employee_feedback`
- The primary key is `review_id`, which determines all other attributes. Since `review_id` is a superkey, this table is in BCNF.

10 Frontend and Backend Implementation

This section outlines the technology stack for developing a complete web application based on our car dealership database system.

10.1 Backend Implementation

Database Management System: PostgreSQL for robust SQL compliance, excellent performance, and ACID compliance ensuring data integrity.

Server-Side Framework: FastAPI (Python) chosen for automatic API documentation, high performance with async support, built-in validation, and easy PostgreSQL integration through SQLAlchemy ORM.

Database Connectivity: SQLAlchemy providing ORM capabilities, connection pooling, migration support with Alembic, and query optimization.

Authentication: JWT with OAuth 2.0 for secure user authentication and role-based authorization (customers, employees, administrators).

10.2 Frontend Implementation

User Interface Framework: React.js for component-based architecture, virtual DOM efficiency, and excellent developer experience.

State Management: Redux Toolkit for centralized state management, predictable updates, and debugging capabilities.

UI Components: Material-UI (MUI) providing consistent Material Design, responsive components, and built-in accessibility features.

HTTP Client: Axios for API communication with request/response interceptors and error handling.

10.3 Key Application Features

- **User Dashboard:** Customer profiles, purchase history, vehicle browsing
- **Inventory Management:** Real-time stock tracking, automated reorder alerts
- **Sales Processing:** Complete purchase workflow from selection to delivery
- **Employee Portal:** Staff management, performance tracking, administrative tools
- **Reporting System:** Business intelligence dashboards with data visualization

- **Mobile Responsiveness:** Optimized experience across all devices

This technology stack ensures scalability, maintainability, and excellent user experience while leveraging our robust database foundation.

11 Conclusion

This project successfully demonstrates the design and implementation of a comprehensive car dealership database management system. Through systematic database design principles and normalization techniques, we have created a robust and efficient system that addresses the complex requirements of modern automotive retail operations.

11.1 Key Achievements

Our database schema encompasses 11 interconnected tables that effectively model the entire car dealership ecosystem, including users, employees, vehicles, inventory management, purchasing workflows, order processing, shipping logistics, and customer feedback systems. The system achieves Boyce-Codd Normal Form (BCNF), ensuring data integrity through elimination of redundancy and prevention of update anomalies.

The implementation includes 16 diverse SQL queries demonstrating comprehensive functionality:

- Various join operations (natural, inner, outer joins) and multi-table relationships
- Advanced subqueries, CTEs, and pattern matching capabilities
- Complete CRUD operations with proper constraint handling
- Business intelligence features for inventory, sales, and customer analysis

11.2 Technical Implementation

The system implements comprehensive data integrity through primary key constraints, foreign key relationships, check constraints, and audit trails. Performance optimization is achieved through proper indexing, normalized structure, and efficient query design, creating a scalable architecture supporting future growth.

11.3 System Benefits

The system streamlines dealership operations by centralizing information, automating processes, and enabling data-driven decision making. Management can leverage real-time analytics for inventory forecasting, customer segmentation, employee performance evaluation, and financial optimization.

11.4 Future Enhancements

The current foundation supports future enhancements including integration with external systems, advanced analytics capabilities, mobile applications, and customer portals for enhanced service delivery.

11.5 Final Remarks

This car dealership database management system represents a comprehensive solution that balances theoretical database design principles with practical business requirements. The systematic approach to normalization, combined with versatile query capabilities and robust data integrity measures, creates a reliable foundation for modern automotive retail operations that can adapt to evolving industry needs.