Ride-Sharing System Report

1. User Management

1.1 Rider Registration

Description: Create a new rider profile in the system.

Table: `riders`

Columns: `rider_id (PK)`, `name`, `email (UNIQUE, NOT NULL)`, `phone`, `profile_pic`, `location`

Business Rules:

- Email must be unique and not null.
- Phone number format should be validated if provided.
- `location` stores the initial or preferred pickup area as VARCHAR(200).

1.2 Driver Registration

Description: Create a new driver profile.

Table: `drivers`

Columns: `driver_id (PK)`, `name`, `email (UNIQUE)`, `phone`, `profile_pic`, `customer_base (NUMERIC(5,2))`, `rating (NUMERIC(3,2) DEFAULT 0)`

Business Rules:

- Email must be unique.
- `customer_base` defaults to 0.00 on creation.
- `rating` defaults to 0.00 and updates as rides are rated.

1.3 Update Profile Information

Description: Allow riders or drivers to update personal details.

Tables: `riders`, `drivers`

Business Rules:

- New email must remain unique.
- Changes should be audited if required (e.g., store history in logs).

2. Vehicle Management

2.1 Car Registration

Description: A driver registers a car for ride services.

Table: `registered_cars`

Columns: `car_id (PK)`, `driver_id (FK)`, `car_number (UNIQUE)`, `color`, `license_plate`

Business Rules:

- `car_number` must be unique.
- `driver_id` must reference an existing `drivers.driver_id`.

2.2 Update Insurance Details

Description: Add or update insurance for a registered car.

Table: `insurance_details`

Columns: `car_id (FK)`, `provider`, `policy_number`, `start_date`, `expiry_date`

Business Rules:

- Composite PK (`car_id`, `start_date`) allows multiple policies over time.
- `car_id` must reference `registered_cars.car_id`.
- Policies with past `expiry_date` should be flagged expired.

3. Ride Management

3.1 Request a Ride

Description: Rider requests a ride, creating a `rides` entry.
Table: `rides`
Columns: `ride_id (PK)`, `rider_id (FK)`, `driver_id (FK)`, `car_id (FK)`, `pickup_location`, `drop_location`, `cost_estimate`
Business Rules:

- `rider_id`, `driver_id`, `car_id` must exist.
- `status` defaults to NULL or set explicitly to 'en_route' / 'pending'.
- `cost_estimate` optional pre-ride.

3.2 Start / Update / Complete Ride

Description: Track ride lifecycle.
Table: `rides` (update columns)
Columns: `start_time`, `end_time`, `status` (CHECK in ('en_route','completed','canceled'))
Business Rules:

- `start_time` set at pickup.
- `end_time` set at completion/cancellation.
- Status transitions enforced by application logic.

3.3 GPS Location Updates

Description: Store location points during rides.
Table: `gps_location_data`
Columns: `ride_id (FK)`, `timestamp (PK)`, `latitude`, `longitude`
Business Rules:

- `ride_id` must exist in `rides`.

4. Financial Operations

4.1 Add / Update Bank Cards

Description: Rider adds or updates payment cards.
Table: `bank_cards`
Columns: `card_id (PK)`, `rider_id (FK)`, `card_number`, `expiry`, `card_type`
Business Rules:

- Store only masked `card_number`.
- Validate `expiry` format (MM/YY).

4.2 Generate Ride Receipt

Description: Generate receipt after ride completion.
Table: `ride_receipts`

Columns: `ride_id (PK, FK)`, `fare`, `tip`, `total`
Business Rules:

- `total = fare + tip`.
- `ride_id` must reference a completed ride.

4.3 Revenue Reporting
Description: Summarize earnings by driver/period.
Derived From: `ride_receipts`, `rides`
Business Rules:

- Group by `driver_id`, date_trunc('month', start_time).

5. Feedback and Support

5.1 Ratings & Reviews
Description: Rider rates driver post-ride.
Table: `rated_in`
Columns: `ride_id (FK)`, `driver_id (FK)`, `rating (1-5)`, `comment`
Business Rules:

- Rating CHECK ensures 1 ≤ rating ≤ 5.
- PK on (ride_id, driver_id) prevents duplicate ratings.

5.2 Complaints
Description: Rider files complaint against driver.
Table: `complaints`
Columns: `complaint_id (PK)`, `rider_id (FK)`, `reporter_id (FK)`, `reported_id (FK)`, `complaint_details`, `date_reported`
Business Rules:

- `reporter_id` and `rider_id` reference `riders`.
- `reported_id` references `drivers`.

5.3 Support Tickets
Description: Rider opens support ticket.
Table: `support_tickets`
Columns: `ticket_id (PK)`, `rider_id (FK)`, `issue`, `status`, `created_at`
Business Rules:

- Status values: 'open', 'in progress', 'resolved'.

6. Reporting and Analytics

6.1 Operational Logs
Description: System logs or summary reports.
Table: *(Dropped)* `reports` removed; use external logging/audit tables.

6.2 Popular Routes & Demand
Description: Compute hotspots from `rides` and `gps_location_data`.

6.3 Driver Availability

Description: Check drivers without active `rides` in last X minutes.

6.4 Financial Summaries

Description: Monthly/quarterly totals via `ride_receipts`.