Relaciones (Foreign Keys)

- trips → vehicles (vehicle_id)

Un vehículo puede tener múltiples viajes.
Un viaje solo puede ser realizado por un vehículo.

- trips → drivers (driver_id)

Un conductor puede realizar múltiples viajes.
Un viaje solo puede tener un conductor.

- trips → routes (route_id)

Una ruta puede ser usada en múltiples viajes.
Un viaje sigue una única ruta predefinida.

- deliveries → trips (trip_id)

Un viaje puede tener múltiples entregas.
Una entrega pertenece a un solo viaje.

- maintenance → vehicles (vehicle_id)

Un vehículo puede tener múltiples mantenimientos.
Cada mantenimiento corresponde a un vehículo específico.

Constraints únicos

- vehicles: license_plate (cada placa es única)
- drivers: employee_code, license_number (códigos únicos)
- routes: route_code (código único de ruta)
- deliveries: tracking_number (número de rastreo único)

Constraints de negocio implícitos

- arrival_datetime ≥ departure_datetime en trips finalizados
- delivered_datetime ≥ scheduled_datetime en deliveries
- next_maintenance_date ≥ maintenance_date
- El peso total del viaje no debe superar la capacidad del vehículo

Justificacion De Tablas

generate_trips()

Genera 100.000 viajes simulados en un período de 2 años.
Antes de crear viajes, obtiene de la base datos válidos para respetar relaciones:
vehículos activos (vehicle_id, capacity_kg)
conductores activos (driver_id)
rutas (route_id, distance_km, estimated_duration_hours)
Para cada viaje:
Selecciona un vehículo, conductor y ruta existentes.
Define departure_datetime usando una hora elegida con _get_hourly_distribution() (más probable en horario laboral).
Calcula arrival_datetime a partir de la duración estimada con una variación realista.
Calcula fuel_consumed_liters en función de la distancia.
Calcula total_weight_kg como porcentaje de la capacidad del vehículo.
Asigna el estado:
completed si el viaje ya debería haber terminado
in_progress si todavía está en curso (llegada NULL)
Inserta los registros en batches para mejorar performance.

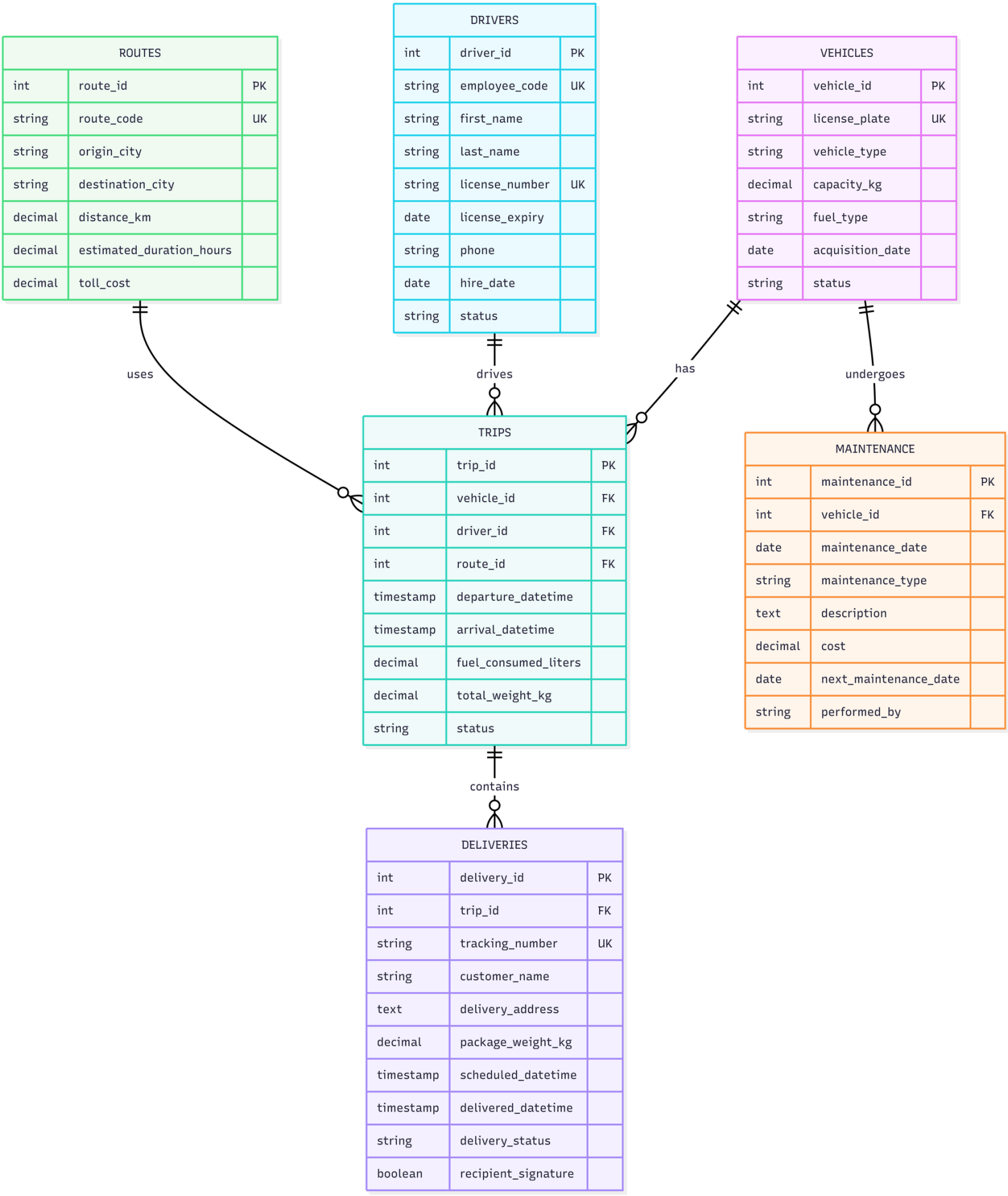
_get_hourly_distribution()

Devuelve una distribución de probabilidades para elegir la hora de salida del viaje.

Aumenta la probabilidad de salidas entre 6:00 y 20:00 (horario operativo).

Define picos de actividad en:
8:00–12:00 (pico mañana)
14:00–18:00 (pico tarde)

Resultado: los viajes no se generan uniformemente en el día, sino con un patrón más realista de operación logística.


ROUTES
int route_id PK
string route_code UK
string origin_city
string destination_city
decimal distance_km
decimal estimated_duration_hours
decimal toll_cost
DRIVERS
int driver_id PK
string employee_code UK
string first_name
string last_name
string license_number UK
date license_expiry
string phone
date hire_date
string status
VEHICLES
int vehicle_id PK
string license_plate UK
string vehicle_type
decimal capacity_kg
string fuel_type
date acquisition_date
string status
uses
drives
has
undergoes
TRIPS
int trip_id PK
int vehicle_id FK
int driver_id FK
int route_id FK
timestamp departure_datetime
timestamp arrival_datetime
decimal fuel_consumed_liters
decimal total_weight_kg
string status
MAINTENANCE
int maintenance_id PK
int vehicle_id FK
date maintenance_date
string maintenance_type
text description
decimal cost
date next_maintenance_date
string performed_by
contains
DELIVERIES
int delivery_id PK
int trip_id FK
string tracking_number UK
string customer_name
text delivery_address
decimal package_weight_kg
timestamp scheduled_datetime
timestamp delivered_datetime
string delivery_status
boolean recipient_signature