

Dark Springs

Jeremy, Gracie, Ashwin, Yoko

February 26, 2025

Dark Springs

Efficiency

At Dark Springs, we **redefine efficiency** by providing seamless hydration solutions, specializing in the timely and reliable delivery of bottled water to various organizations.

Value prop

At Dark Springs, we provide **reliable, hassle-free water delivery services** that keep businesses, schools, and healthcare facilities hydrated and operational.

Messaging

We have the expertise, scale and reach to make an impact and approach opportunities to advance efficiency through:

- **Optimizing Operations:** Streamlining transactions, enforcing business rules, automating financial processes, and ensuring accurate record-keeping.
- **Leveraging Data & Insights:** Providing actionable analytics that empower our partners with tailored solutions and strategic recommendations, improving access and operational effectiveness.

Functionality: Why a Robust Database Matters



Streamlining Operations & Transactions

- Automates contract management, ensuring all agreements dynamically update for their 1-year duration.
- Facilitates seamless order processing, tracking deliveries, sales, and inventory in real-time



Business Rules & Compliance

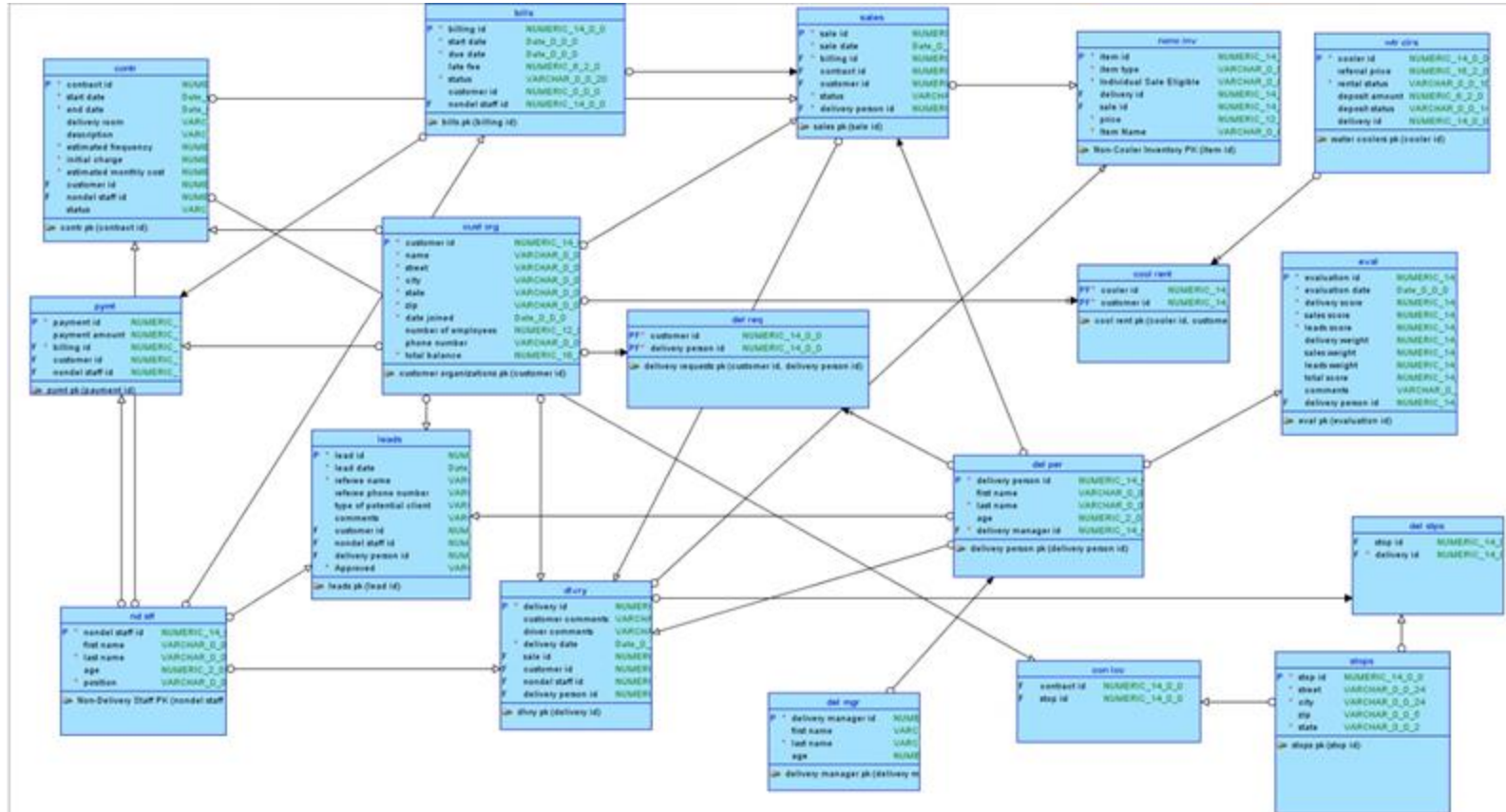
- Automatically calculates and applies late fees, deposit deductions, and billing adjustments based on defined rules.
- Maintains data integrity with constraints on billing status, contract terms, and evaluation scoring.



Enhancing Financial Accuracy & Automation

- Automates invoicing on the 1st of every month, prorating charges based on service start dates.
- Handles multi-payment billing, ensuring correct allocations and late fee enforcement.
- Processes referral discounts dynamically, reflecting them in customer invoices.

Database Structure Overview



Business Rules and Enforcement

Billing & Payment Rules

Rule: Late fees are automatically applied (10% of the outstanding amount) if full payment isn't received by the due date.

Enforced by: A scheduled job that checks overdue balances and adds late fees.

Rule: Approved leads generate a \$25 discount on the customer's billing.

Enforced by: A trigger that automatically applies the discount when a lead is approved.

Rule: Monthly billing is generated on the 1st of the next month and is prorated by days of service.

Enforced by: A scheduled job that calculates and issues invoices based on contract start dates.

Rule: Customers can make partial payments, but each payment must be linked to a specific billing cycle.

Enforced by: A one-to-many relationship between payments and billing records.

Contract Management

Rule: Each customer must have an active contract before making a purchase.

Enforced by: Database constraint linking sales to an active contract.

Rule: Contracts last exactly one year, with the end date dynamically set.

Enforced by: A trigger that auto-updates the contract's expiration date based on the start date.

Rule: A contract can cover multiple delivery locations, but each location must be tied to a specific contract.

Enforced by: A one-to-many relationship between contracts and locations.

Sales & Delivery Rules

Rule: Customers must rent a cooler before purchasing large water bottles.

Enforced by: A validation check preventing water bottle purchases without an active cooler rental.

Rule: Deposits for cooler rentals are automatically deducted and refunded upon return.

Enforced by: A trigger that adjusts the customer's balance upon rental and return.

Rule: Estimated delivery frequency determines whether a deposit is waived.

Enforced by: A condition that checks frequency before applying the deposit charge.

Rule: Each delivery must be assigned to one and only one delivery person but can contain multiple stops and items.

Enforced by: Foreign key constraints ensuring deliveries are correctly assigned.

Employee Evaluation & Performance Rules

Rule: Delivery managers evaluate drivers weekly based on the number of deliveries completed.

Enforced by: A validation requiring weekly entries.

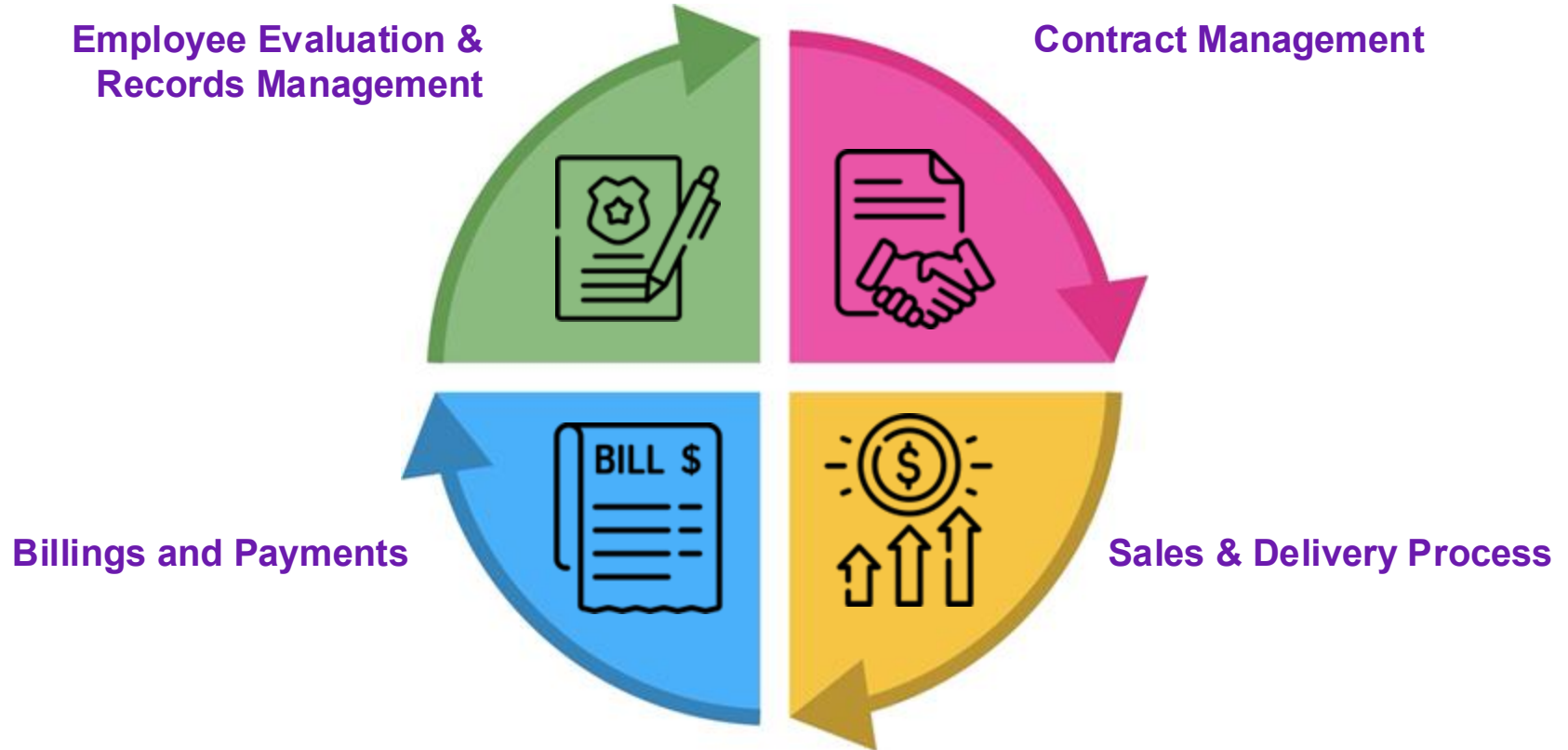
Rule: The average evaluation score is calculated over a set period to determine bonuses and promotions.

Enforced by: A stored procedure that calculates performance scores.

Rule: Evaluation records must be removed 3 years after the driver leaves.

Enforced by: A deletion trigger that purges records based on termination dates

Core Functionalities



Contract Management



Contract Package Allows Users To:

- 1. Find Expiring Contracts
- 1. Renew Existing Contracts
- 1. Create New Contracts
- 1. Deactivate/Terminate Old Contracts

Locate Expiring Contracts in Less Than 30 Days

	CONTRAC...	START_DATE	END_DATE	DELIVERY_ROOM	DESCRIPTION	ESTIMATED_FREQUENCY	INITIAL_CHARGE	ESTIMATED_MONTHLY_COST	CUSTOMER_ID	NONDEL_STAFF_ID	STATUS
1	18	15-MAR-24	15-MAR-25	Room 215	Premium plan	4	510	700.6	6	5	Active
2	17	05-MAR-24	05-MAR-25	Room 102	Standard plan	2	320.5	500.25	1	2	Active

```
SET SERVEROUTPUT ON;
```

```
EXEC contract_actions.find_expiring_contracts;
```

```
Contract ID: 17, Customer: Apple Corp, Days until expiration: 6
```

```
Contract ID: 18, Customer: Fig Logistics, Days until expiration: 16
```


Renew Existing Contracts

CONTRAC...	START_DATE	END_DATE	DELIVERY_ROOM	DESCRIPTION	ESTIMATED_FREQUENCY	INITIAL_CHARGE	ESTIMATED_MONTHLY_COST	CUSTOMER_ID	NONDEL_STAFF_ID	STATUS
1	18 15-MAR-24	15-MAR-25	Room 215	Premium plan	4	510	700.6	6	5	Active
2	17 05-MAR-24	05-MAR-25	Room 102	Standard plan	2	320.5	500.25	1	2	Active

```
EXEC contract_actions.renew_contract(1);
```

Takes a customer id, creates a new 1-year contract based on today's date, retains all other previous information, flips the old contract's status from active to inactive.

CONTRACT_ID	START_DATE	END_DATE	DELIVERY_ROOM	DESCRIPTION	ESTIMATED_FREQUENCY	INITIAL_CHARGE	ESTIMATED_MONTHLY_COST	CUSTOMER_ID	NONDEL_STAFF_ID	STATUS
1	19 27-FEB-25	27-FEB-26	Room 102	Standard plan	2	320.5	500.25	1	2	Active
2	18 15-MAR-24	15-MAR-25	Room 215	Premium plan	4	510	700.6	6	5	Active
3	17 05-MAR-24	05-MAR-25	Room 102	Standard plan	2	320.5	500.25	1	2	Inactive

Create New Contracts

CONTRACT_ID	START_DATE	END_DATE	DELIVERY_ROOM	DESCRIPTION	ESTIMATED_FREQUENCY	INITIAL_CHARGE	ESTIMATED_MONTHLY_COST	CUSTOMER_ID	NONDEL_STAFF_ID	STATUS
1	18-15-MAR-24	15-MAR-25	Room 215	Premium plan	4	510	700.6	6	5	Active
2	17-05-MAR-24	05-MAR-25	Room 102	Standard plan	2	320.5	500.25	1	2	Active

```
BEGIN
  contract_actions.create_new_contract(
    10, -- customer_id
    'Room 884', -- delivery_room
    'Standard Plan', -- description
    3, -- estimated_frequency
    190.75, -- initial_charge
    1900.50, -- estimated_monthly_cost
    2 -- nondel_staff_id
  );
END;
```

Using contract_actions.create_new_contract

CONTRACT_ID	START_DATE	END_DATE	DELIVERY_ROOM	DESCRIPTION	ESTIMATED_FREQUENCY	INITIAL_CHARGE	ESTIMATED_MONTHLY_COST	CUSTOMER_ID	NONDEL_STAFF_ID	STATUS
1	19-27-FEB-25	27-FEB-26	Room 884	Standard Plan	3	190.75	1900.5	10	2	Active
2	18-15-MAR-24	15-MAR-25	Room 215	Premium plan	4	510	700.6	6	5	Inactive
3	17-05-MAR-24	05-MAR-25	Room 102	Standard plan	2	320.5	500.25	1	2	Inactive

Terminate a Contract

	CONTRACT_ID	START_DATE	END_DATE	DELIVERY_ROOM	DESCRIPTION	ESTIMATED_FREQUENCY	INITIAL_CHARGE	ESTIMATED_MONTHLY_COST	CUSTOMER_ID	NONDEL_STAFF_ID	STATUS
1	19 27-FEB-25	27-FEB-26	Room 884	Standard Plan	3	190.75	1900.5	10	2	Active	
2	18 15-MAR-24	15-MAR-25	Room 215	Premium plan	4	510	700.6	6	5	Inactive	
3	17 05-MAR-24	05-MAR-25	Room 102	Standard plan	2	320.5	500.25	1	2	Inactive	

Flips the status on customer 10's contract from "Active" to "Inactive"

```
EXEC contract_actions.terminate_contracts(10);
```

	CONTRACT_ID	START_DATE	END_DATE	DELIVERY_ROOM	DESCRIPTION	ESTIMATED_FREQUENCY	INITIAL_CHARGE	ESTIMATED_MONTHLY_COST	CUSTOMER_ID	NONDEL_STAFF_ID	STATUS
1	19 27-FEB-25	27-FEB-26	Room 884	Standard Plan	3	190.75	1900.5	10	2	Inactive	
2	18 15-MAR-24	15-MAR-25	Room 215	Premium plan	4	510	700.6	6	5	Inactive	
3	17 05-MAR-24	05-MAR-25	Room 102	Standard plan	2	320.5	500.25	1	2	Inactive	

Sales and Delivery Process



Sales Package Allows Sales Employees To:

1. Create New Sales
1. Update Existing Sales

Creating a New Sale

Step 1: Creates a new row in the sales table

OUTPUT:

SALE_ID	SALE_DATE	BILLING_ID	CONTRACT_ID	CUSTOMER_ID	STATUS	DELIVERY_PERSON_ID
1	18-27-FEB-25	2	2	2	Awaiting Approval	3
2	17-26-FEB-25	1	2	3	Awaiting Approval	1

```

BEGIN
  create_new_sale(
    v_billing => 2, -- Billing ID
    v_contract => 2, -- Contract ID
    v_customer => 2, -- Customer ID
    v_delivery_person => 3, -- Delivery Person ID
    v_inventory_1 => 'Cheap Small Water', -- Item 1
    v_inventory_1_quantity => 1, -- Item 1 Quantity
    v_inventory_2 => 'Small Cup', -- Item 2
    v_inventory_2_quantity => 1 -- Item 2 Quantity
  );
END;
/

```

Updates the inventory table to reflect that those items are part of the new sale

ITEM_ID	ITEM_TYPE	Individual_Sale_Eligible	DELIVERY_ID	SALE_ID	PRICE	ITEM_NAME
1	18 Miscellaneous	No	6	5	20	5-Pack Large Cups
2	17 Miscellaneous	Yes	(null)	(null)	5.5	Small Cup
3	16 Large Water Bottle	No	5	2	4.5	Large Water Bottle
4	15 Small Water Bottle	Yes	(null)	(null)	2	Cheap Small Water

ITEM_ID	ITEM_TYPE	Individual_Sale_Eligible	DELIVERY_ID	SALE_ID	PRICE	ITEM_NAME
1	18 Miscellaneous	No	6	5	20	5-Pack Large Cups
2	17 Miscellaneous	Yes	(null)	18	5.5	Small Cup
3	16 Large Water Bottle	No	5	2	4.5	Large Water Bottle
4	15 Small Water Bottle	Yes	(null)	18	2	Cheap Small Water

Change Existing Sales

1	SALE_ID	SALE_DATE	BILLING_ID	CONTRACT_ID	CUSTOMER_ID	STATUS	DELIVERY_PERSON_ID
1	18	27-FEB-25	2	2	2	Requesting Update	3
2	17	26-FEB-25	1	2	3	Awaiting Approval	

```

BEGIN
  sales_actions.update_sale(
    v_sale_id => 18,
    v_old_product => 'Cheap Small Water',
    v_new_product => 'Large Water Bottle'
  );
END;

```

1	SALE_ID	SALE_DATE	BILLING_ID	CONTRACT_ID	CUSTOMER_ID	STATUS	DELIVERY_PERSON_ID
1	18	27-FEB-25	2	2	2	Approved	3
2	17	26-FEB-25	1	2	3	Awaiting Approval	1

1	ITEM_ID	ITEM_TYPE	Individual_Sale_Eligible	DELIVERY_ID	SALE_ID	PRICE	ITEM_NAME
1	19	Large Water Bottle	No	(null)	(null)	4.5	Large Water Bottle
2	18	Miscellaneous	No	6	5	20	5-Pack Large Cups
3	17	Miscellaneous	Yes	(null)	18	5.5	Small Cup

Changed the
item_name

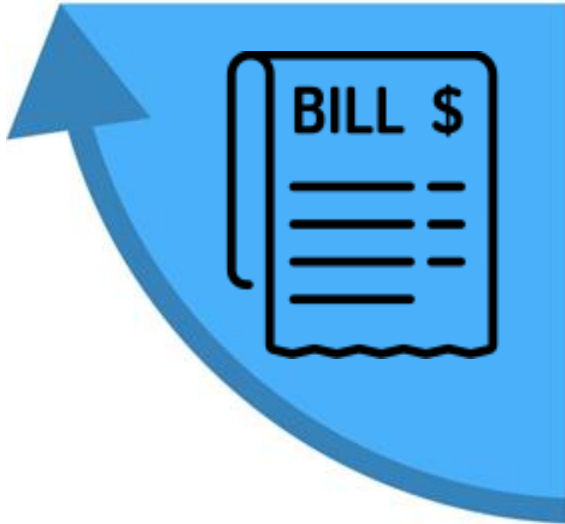
```

v_old_product => 'Cheap Small Water',
v_new_product => 'Large Water Bottle'

```

1	ITEM_ID	ITEM_TYPE	Individual_Sale_Eligible	DELIVERY_ID	SALE_ID	PRICE	ITEM_NAME
1	19	Large Water Bottle	No	(null)	18	4.5	Large Water Bottle
2	18	Miscellaneous	No	6	5	20	5-Pack Large Cups
3	17	Miscellaneous	Yes	(null)	18	5.5	Small Cup

Billing and Payments



New Functions Offer Automatic Changes in Billings Table:

1. Automatic late fee application (10%) for overdue payments.
2. Implemented an automated trigger that applies a \$25 discount to a customer organization's balance whenever a lead from that organization is approve
3. Active bills are calculated and automatically sent to customers on the first day of each month.

Calculate and Apply Late Fees

	BILLING_ID	START_DATE	DUE_DATE	LATE_FEE	 ¹ STATUS	CUSTOMER_ID	NONDEL_STAFF_ID
1	9	01-JAN-25	01-FEB-25	0	Awaiting Payment	9	(null)
2	3	01-JUN-25	01-JUL-25	0	Awaiting Payment	3	(null)
3	7	01-DEC-24	01-JAN-24	0	Awaiting Payment	7	(null)
4	1	01-FEB-24	01-MAR-24	0	Awaiting Payment	1	(null)

```
EXEC update_late_fees();
```

Late Fee = 10% of Original Bill *
Number of Months Late

	BILLING_ID	TOTAL_COST	MONTHS_LATE	LATE_FEE	TOTAL_LATE_FEE
1	9	105	1	10.5	10.5
2	1	4	11	0.4	4.4
3	7	10	13	1	13

	BILLING_ID	START_DATE	DUE_DATE	LATE_FEE	 ¹ STATUS	CUSTOMER_ID	NONDEL_STAFF_ID
	9	01-JAN-25	01-FEB-25	10.5	Awaiting Payment	9	(null)
	3	01-JUN-25	01-JUL-25	0	Awaiting Payment	3	(null)
	7	01-DEC-24	01-JAN-24	13	Awaiting Payment	7	(null)
	1	01-FEB-24	01-MAR-24	4.4	Awaiting Payment	1	(null)

Calculate and Send Active Bills

```
SET SERVEROUTPUT ON;
EXEC Calc_Send_Billing();
```

For all active bills, calculate the cost of sales associated with that bill and apply it to the customer's total balance. This is set as a scheduled job on the first of each month.

BILLI...	START_DATE	DUE_DATE	LATE_FEE	STATUS	CUSTOMER_ID	NONDEL_STAFF_ID
1	9 01-JAN-25	01-FEB-25	10.5	Active	9	(null)
2	3 01-JUN-25	01-JUL-25	0	Active	3	(null)
3	7 01-DEC-24	01-JAN-24	0	Active	7	(null)
4	1 01-FEB-24	01-MAR-24	4.8	Active	1	(null)
5	8 01-NOV-24	01-DEC-24	0	Paid	8	(null)

BILLING_ID	START_DATE	DUE_DATE	LATE_FEE	STATUS	CUSTOMER_ID	NONDEL_STAFF_ID
1	9 01-JAN-25	01-FEB-25	10.5	Awaiting Payment	9	(null)
2	3 01-JUN-25	01-JUL-25	0	Awaiting Payment	3	(null)
3	7 01-DEC-24	01-JAN-24	0	Awaiting Payment	7	(null)
4	1 01-FEB-24	01-MAR-24	4.8	Awaiting Payment	1	(null)

CUSTOMER_ID	NAME	STREET	CITY	STATE	ZIP	DATE_JOINED	NUMBER_OF_EMPLOYEES	PHONE_NUMBER	TOTAL_BALANCE
1	1 Apple Corp	123 Main St	Albuquerque NM	87101	15-MAY-20		150	505-555-1234	0
2	3 Cherry Solutions	789 Oak St	Albuquerque NM	87103	11-JUL-19		200	505-555-9876	0
3	9 Kiwi Solutions	606 Spruce St	Albuquerque NM	87109	14-MAR-15		45	505-555-1122	0
4	7 Grape Manufacturing	404 Walnut St	Albuquerque NM	87107	05-NOV-16		350	505-555-8642	0

CUSTOMER_ID	NAME	STREET	CITY	STATE	ZIP	DATE_JOINED	NUMBER_OF_EMPLOYEES	PHONE_NUMBER	TOTAL_BALANCE
1	9 Kiwi Solutions	606 Spruce St	Albuquerque NM	87109	14-MAR-15		45	505-555-1122	-115.5
2	1 Apple Corp	123 Main St	Albuquerque NM	87101	15-MAY-20		150	505-555-1234	-68.8
3	7 Grape Manufacturing	404 Walnut St	Albuquerque NM	87107	05-NOV-16		350	505-555-8642	-10
4	3 Cherry Solutions	789 Oak St	Albuquerque NM	87103	11-JUL-19		200	505-555-9876	-4

Employee Evaluation & Records Management



New Functions Offer Automatic Changes in Employee Evaluation Table:

1. Weekly evaluations of drivers based on deliveries and leads.
2. Performance scores impact bonuses & promotions.
3. Driver evaluations automatically deleted after 3 years post-exit

Employee Evaluation Function

```
VARIABLE rc REFCURSOR;  
BEGIN  
    :rc := get_delivery_person_stats(1, 12, 2024);  
END;  
/  
PRINT rc;
```

Delivery Person ID

Month of Interest

Year of Interest

Can be used by the delivery manager to quickly see the stats of one of their delivery people as they fill out their monthly evaluation.

Revenue from Deliveries TOTAL_DELIVERIES TOTAL_SALES LEADS

4 1 1 1

Roles

ROLE	TABLE_NAME	PRIVILEGE
DELIVERY_MANAGER_EVAL	EVAL	UPDATE
DELIVERY_MANAGER_EVAL	EVAL	DELETE
DELIVERY_MANAGER_EVAL	EVAL	SELECT
DELIVERY_MANAGER_EVAL	EVAL	INSERT

ROLE	TABLE_NAME	PRIVILEGE
CHRO	EVAL	SELECT

Delivery Manager Evaluation Role: Allows delivery managers to view and modify the EVAL table. It enables managers to add and review evaluation scores and comments for delivery personnel.

CHRO Role: Grants the Chief HR Officer view-only access to the EVAL table. It helps HR assess delivery personnel performance and make strategic decisions to boost productivity.

Alternate index

INDEX_NAME	TABLE_NAME	UNIQUENES
DELIVERY_MANAGER_PK	DEL_MGR	UNIQUE
IDX_DELIVERY_EMPLOYEE	DEL_MGR	NONUNIQUE
DELIVERY_PERSON_PK	DEL_PER	UNIQUE
IDX_DELIVERY_PERSON	DEL_PER	NONUNIQUE

Unique Indexes: **DELIVERY_MANAGER_PK** (on **DEL_MGR**) and **DELIVERY_PERSON_PK** (on **DEL_PER**) ensure that each delivery manager and delivery person has a unique identifier, preventing duplicate entries and maintaining data integrity.

Non - Unique Indexes: **IDX_DELIVERY_EMPLOYEE** (on **DEL_MGR**) and **IDX_DELIVERY_PERSON** (on **DEL_PER**) index the first name and last name fields, allowing for faster name-based searches. These indexes improve query efficiency when filtering or sorting by employee names, especially in large datasets where sequential scans would be inefficient

Denormalization

DELIVERY_P...	DELIVERY_P...	DEL_PERS...	DELIVERY_MAN...	MANAGER
4	Ava Rodriguez	33		4 Ashwin Patel
1	Zendaya Her...	35		1 Gracie Anderson
5	Maria Lopez	40		5 Jeremy Clark
3	Olivia Mart...	29		3 Jewel Martinez
9	Rick Ortega	45		9 Karen Bennett
6	Bob Sanchez	27		6 Kris Wong
8	Matthew Torres	31		8 Shambu Gupta
7	Thomas Ramirez	38		7 Sharon Hernandez
10	Greg Vazquez	36		10 Tori Miller
2	Tom Garcia	42		2 Yoko Tanaka

Denormalization for Improved Efficiency

- **Reduces Costly Joins:** Eliminates frequent joins between DEL_PER and DEL_MGR tables, improving query performance.
- **Optimized for Delivery Operations:** Since delivery involves multiple stakeholders, faster data retrieval is crucial.
- **Precomputed Data:** The new DEL_PER_MGR table stores the manager's name alongside delivery personnel details, removing the need for joins.
- **Performance Boost:** Enhances read performance, reduces database load, and speeds up reporting.

Generated Report

Report 1: Awaiting Bill Status

	NAME	CUSTOMER_ID	LATE_FEE	BILLING_ID
1	Apple Corp	1	0	1
2	Grape Manufacturing	7	0	7
3	Kiwi Solutions	9	0	9

Identifies customers with **outstanding payments**, enabling the business to send reminders and improve cash flow.

Report 2: High Performance Delivery Persons

	NAME	DELIVERY_M...	TOTAL_SCORE	COMMENTS
1	Zendaya Hernandez	1	8	Strong delivery performance
2	Tom Garcia	2	8	Outstanding performance
3	Maria Lopez	5	9	Excellent performance
4	Thomas Ramirez	7	8	Very good performance

Highlights **top-performing delivery staff**, allowing management to offer incentives, boost morale, and drive overall performance

Data Warehouse Migration Guide

Table Name	Type	Description
ORDERS_FACT	Fact	Stores the total number of orders, which is the measurable metric in the data warehouse
ZIP_CODE_DIM	Dimension	Represents customer locations, extracted from the CUST_ORG table. Helps analyze orders by geographic region
ITEM_TYPE_DIM	Dimension	Categorizes different item types, sourced from the CONC_INV table. Useful for tracking sales trends by product category
DESCRIPTION_DIM	Dimension	Contains product descriptions from the CONTR table. Provides details about packaging of items being sold