https://www.linkedin.com/in/ahamedafsal/

Case Study

Lawn Squad: Designing Lawn Care Services

Prepared by: Ahamed Afsal

https://www.linkedin.com/in/ahamedafsal/

Table of Contents

https://www.linkedin.com/in/ahamedafsal/

1. About Lawn Squad

Lawn Squad is a Calgary-based service company dedicated to delivering high-quality lawn care and maintenance solutions to both **residential** and **commercial** clients. With a growing customer base, the company provides a variety of seasonal and year-round services aimed at enhancing outdoor spaces.

As part of its digital transformation initiative, Lawn Squad is moving away from manual record-keeping to a **data-driven system** that improves service delivery, scheduling, customer management, and performance analysis. The project focuses on designing and implementing a structured database system that supports business operations, enhances customer experience, and enables future scalability.

Services Offered:

- ♦ Lawn Aeration
- ♦ Power Raking
- ♦ Window Cleaning
- ♦ Fertilization

https://www.linkedin.com/in/ahamedafsal/

2. Mission and Objectives

Mission:

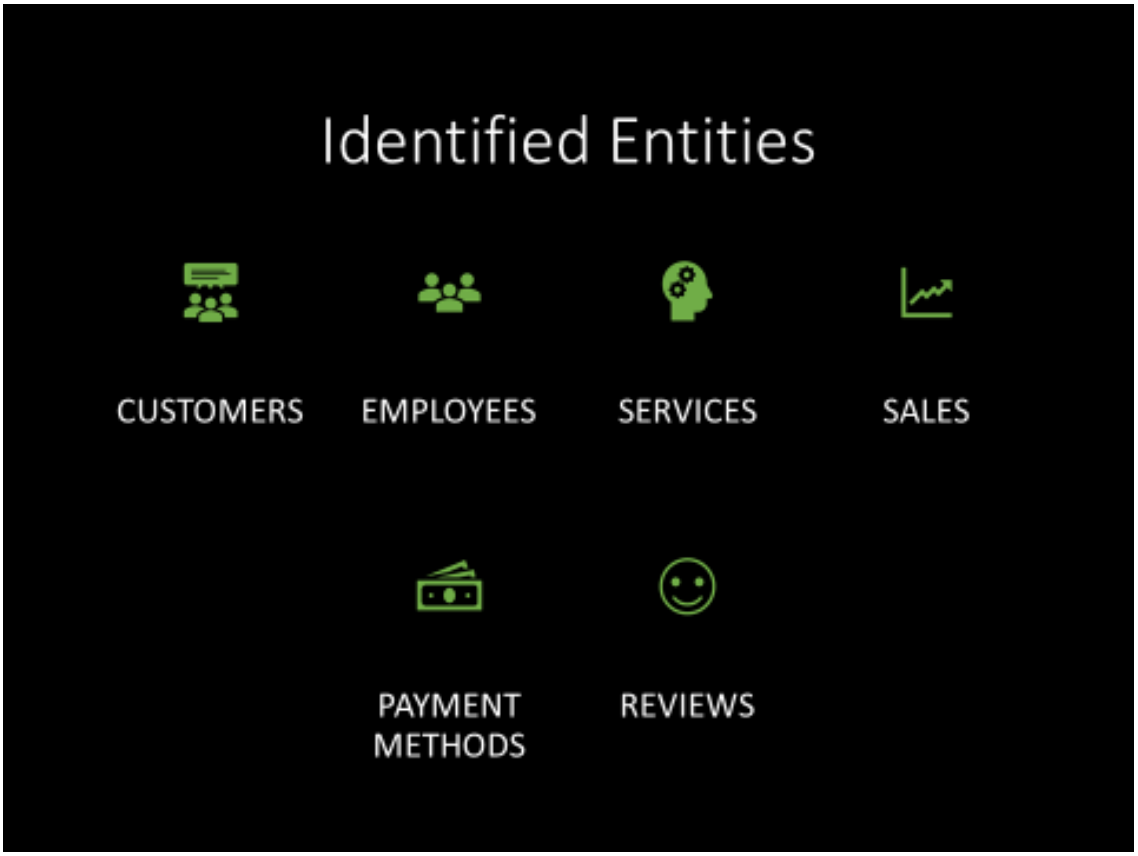
“To design and implement a digital data system for Lawns Squad.”

Objectives:

- Streamline service management.
- Implement a digital database system.
- Enable data-driven decision making.

https://www.linkedin.com/in/ahamedafsal/

3. Identified Entities



https://www.linkedin.com/in/ahamedafsal/

4. Entity Relationships

Definition:
Understanding how the entities relate to one another is crucial for building a normalized and relational database. Below are the key relationships used in Lawn Squad's database design:

- **Employees → Sales** : One-to-Many
- **Customers → Sales** : One-to-Many
- **Sales → Services** : Many-to-One
- **Sales → Payment Methods**: Many-to-One
- **Customers → Reviews** : One-to-Many

Employees Sales: One-to-Many

Definition: One employee can be associated with multiple sales, but each sale is linked to only one employee.
Why it matters: Tracks employee workload and performance based on assigned services.
Example: Alex (employee) is assigned to 5 different customer jobs—each of those sales is connected to Alex.

Customers Sales: One-to-Many

Definition: A single customer can request multiple services over time, resulting in multiple sales records.

Why it matters: Helps maintain a history of customer interactions and supports repeat service tracking.

Example: John Doe (customer) books lawn aeration in April and window cleaning in June—both sales link back to John.

Sales Services: Many-to-One

Definition: Many sales can be associated with a single type of service, but each sale pertains to only one service.

Why it matters: Simplifies service classification and reporting (e.g., most requested services).

Example: 20 sales records for "Fertilization" all point to the same "Fertilization" entry in the Services table.

Sales Payment Methods: Many-to-One

Definition: Each sale uses one payment method, but one payment method can be used in many sales.

Why it matters: Enables analysis of how customers prefer to pay (credit, cash, e-transfer, etc.).

Example: Out of 100 sales, 65 were paid using credit card—all referencing the same "Credit Card" method.

https://www.linkedin.com/in/ahamedafsal/

Customers Reviews: One-to-Many

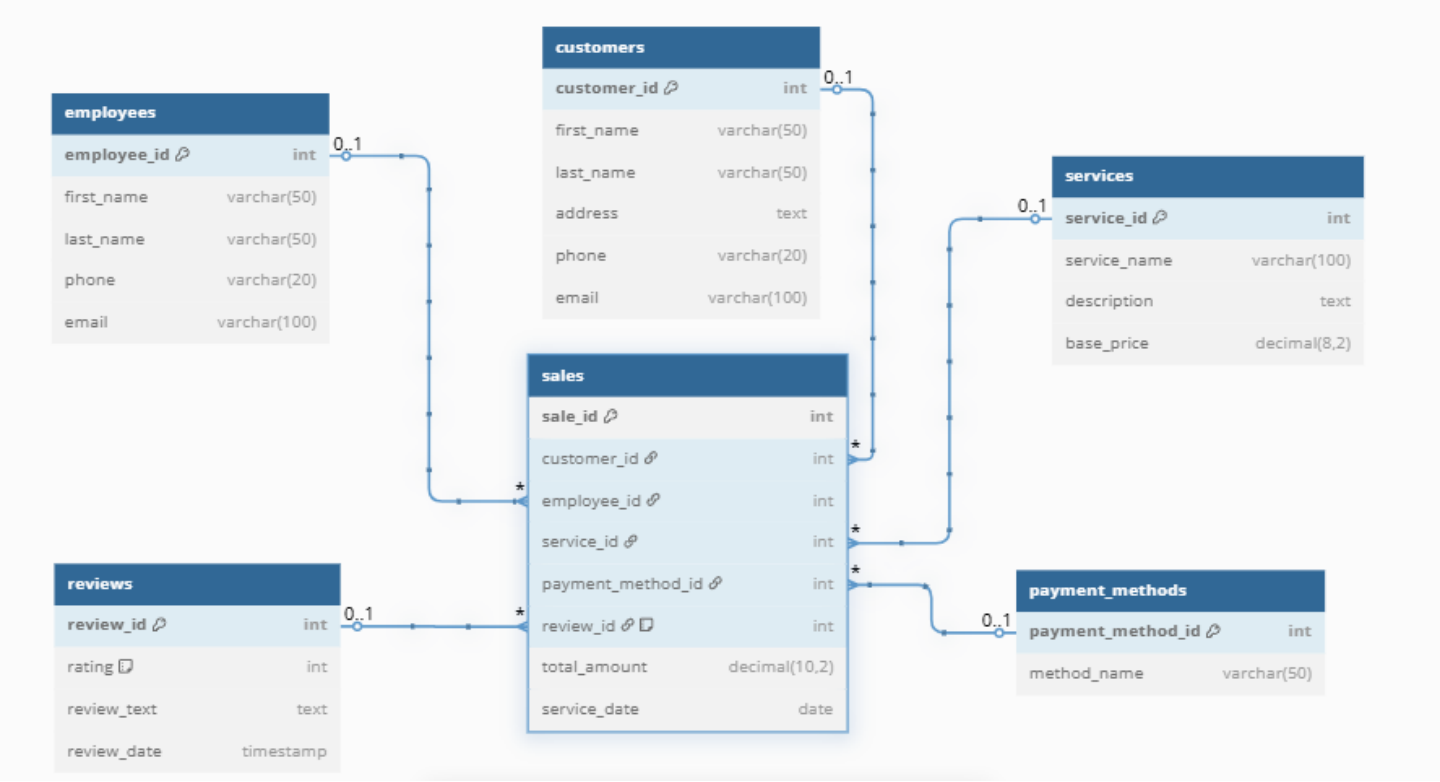
Definition: One customer may leave multiple reviews, each connected to a different service experience.

Why it matters: Supports quality assurance and service improvement through feedback tracking.

Example: Sarah (customer) leaves a review after lawn aeration and another after window cleaning—both reviews are tied to her profile.

5. Entity Relationship Diagram

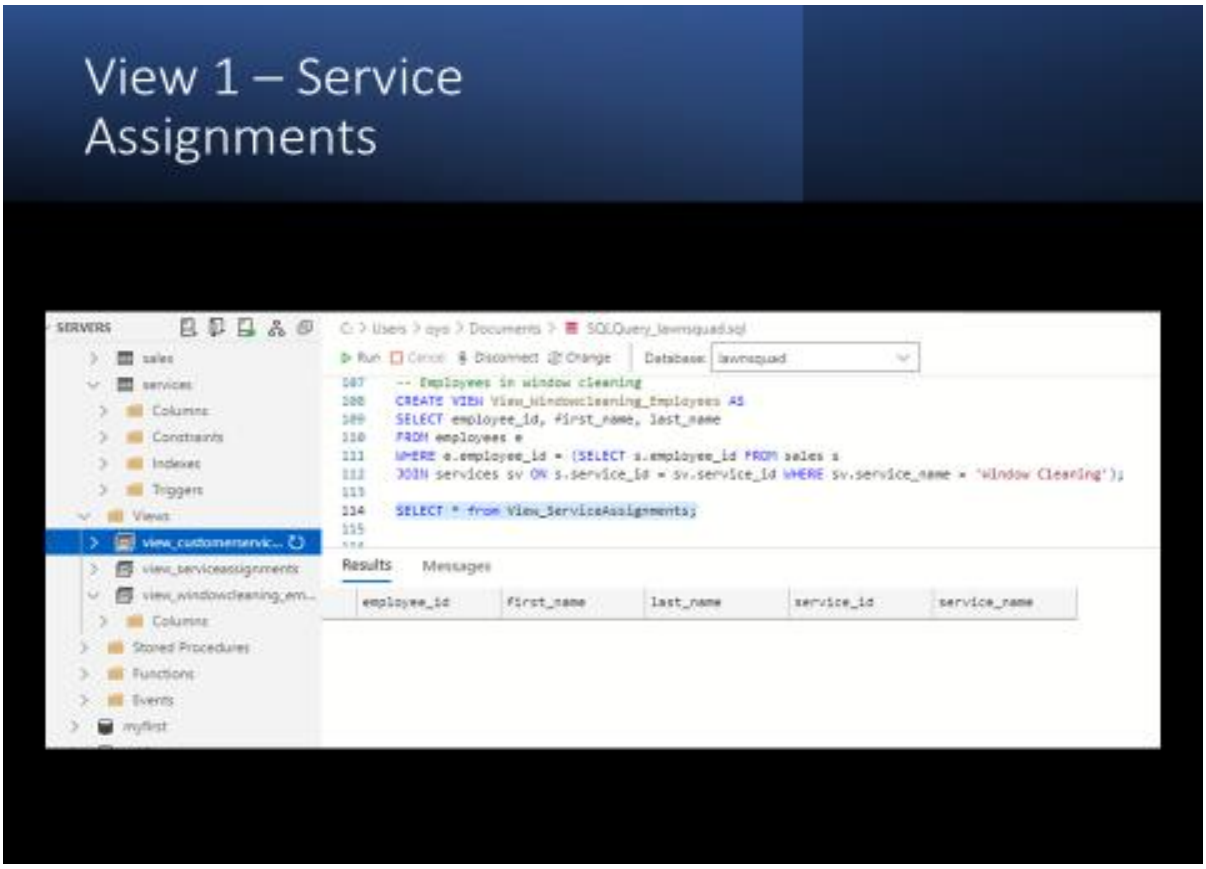
Refer to the attached ER diagram slide in the original presentation.



https://www.linkedin.com/in/ahamedafsal/

6. Data Views

View 1 – Service Assignments

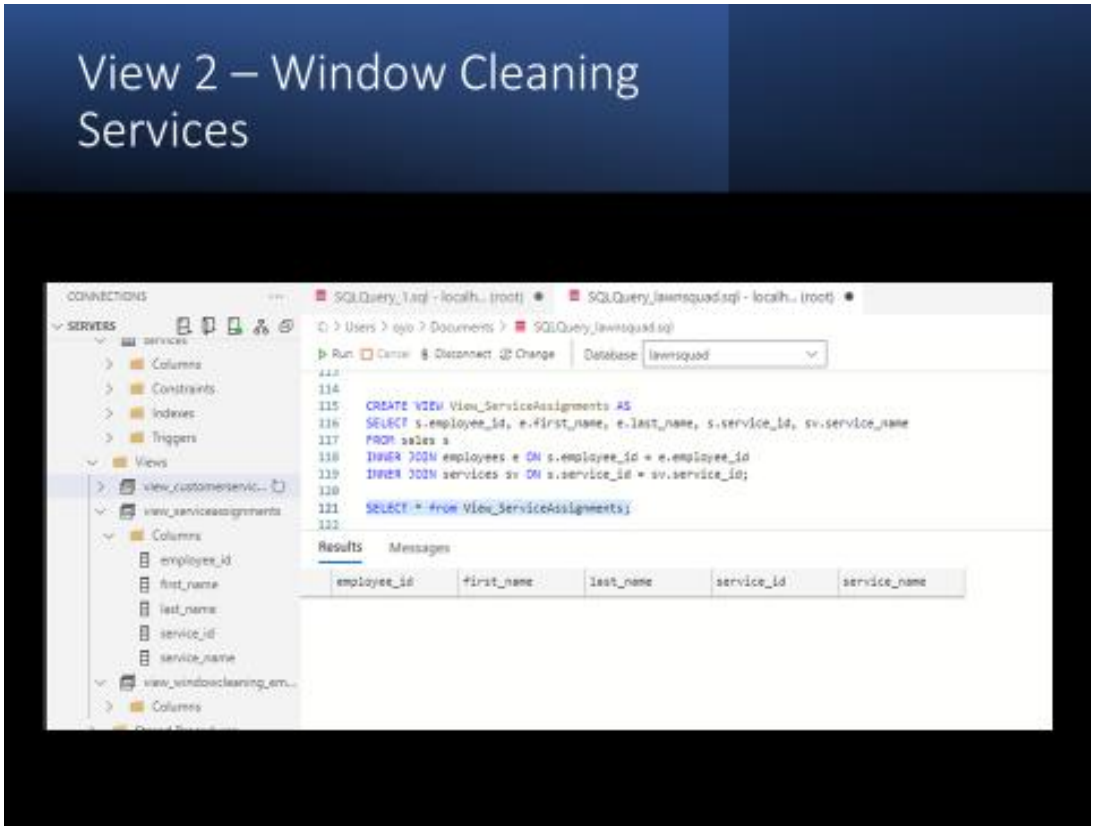


SQL QUERIES

```
CREATE VIEW View_ServiceAssignments AS
SELECT s.employee_id, e.first_name, e.last_name,
s.service_id, sv.service_name
FROM sales s
INNER JOIN employees e ON s.employee_id =
e.employee_id
INNER JOIN services sv ON s.service_id =
sv.service_id;
```

https://www.linkedin.com/in/ahamedafsal/

View 2 – Window Cleaning Services

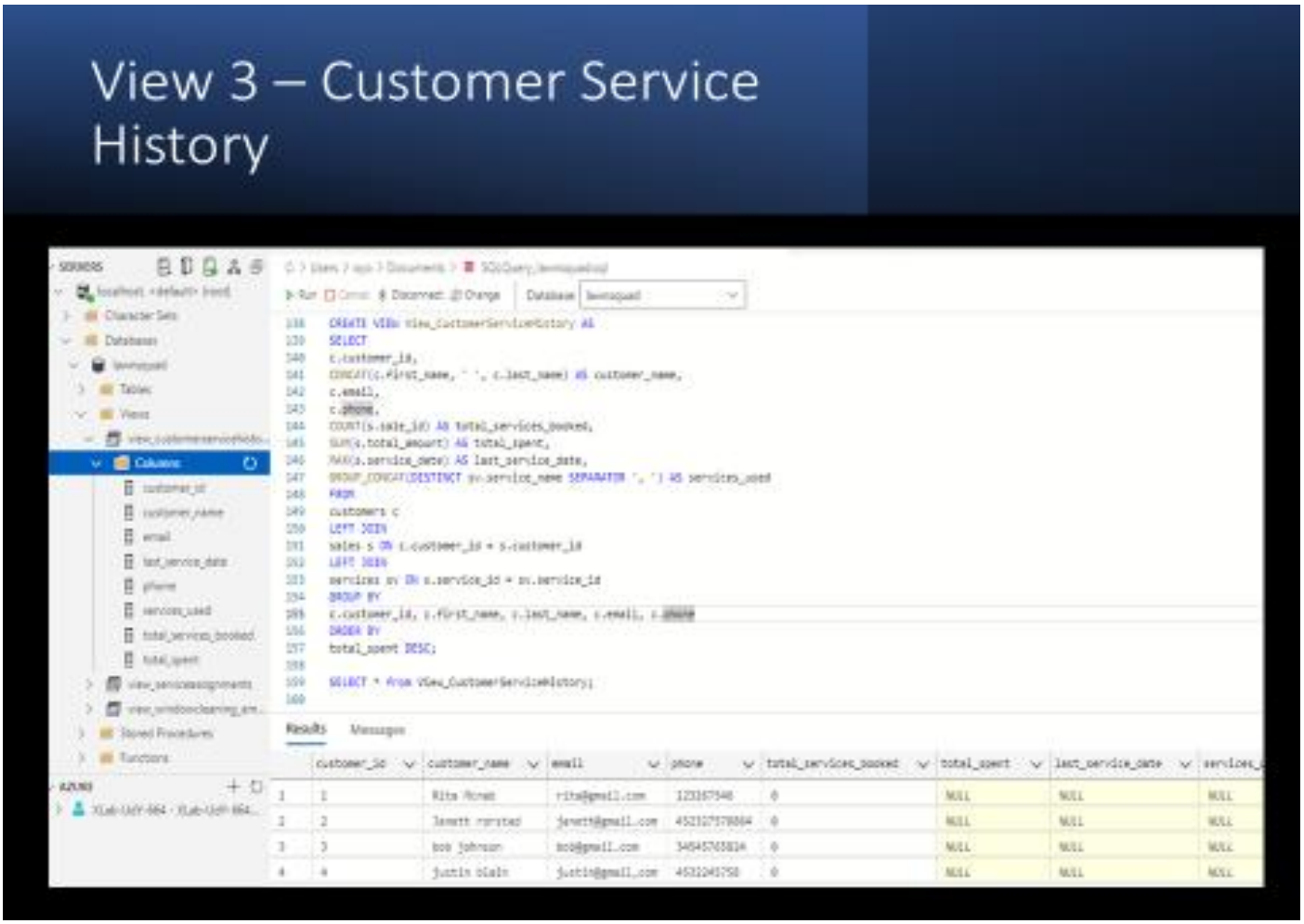


SQL QUERIES

```
-- Employees in window cleaning
CREATE VIEW View_Windowcleaning_Employees AS
SELECT employee_id, first_name, last_name
FROM employees e
WHERE e.employee_id = (SELECT s.employee_id FROM
sales s
JOIN services sv ON s.service_id = sv.service_id
WHERE sv.service_name = 'Window Cleaning');
```

https://www.linkedin.com/in/ahamedafsal/

View 3 – Customer Service History



SQL QUERIES

```
SELECT
    c.customer_name,
    s.service_name,
    r.review_text,
    r.rating,
    r.review_date
FROM
    Reviews r
JOIN Customers c ON r.customer_id = c.customer_id
```

https://www.linkedin.com/in/ahamedafsal/

```
JOIN Sales sa ON r.sale_id = sa.sale_id
JOIN Services s ON sa.service_id = s.service_id
ORDER BY
    r.review_date DESC;
```

7. Conclusion & Acknowledgment

Thank you for reviewing our project. We look forward to implementing the next stages of designing data for Lawn Squad.