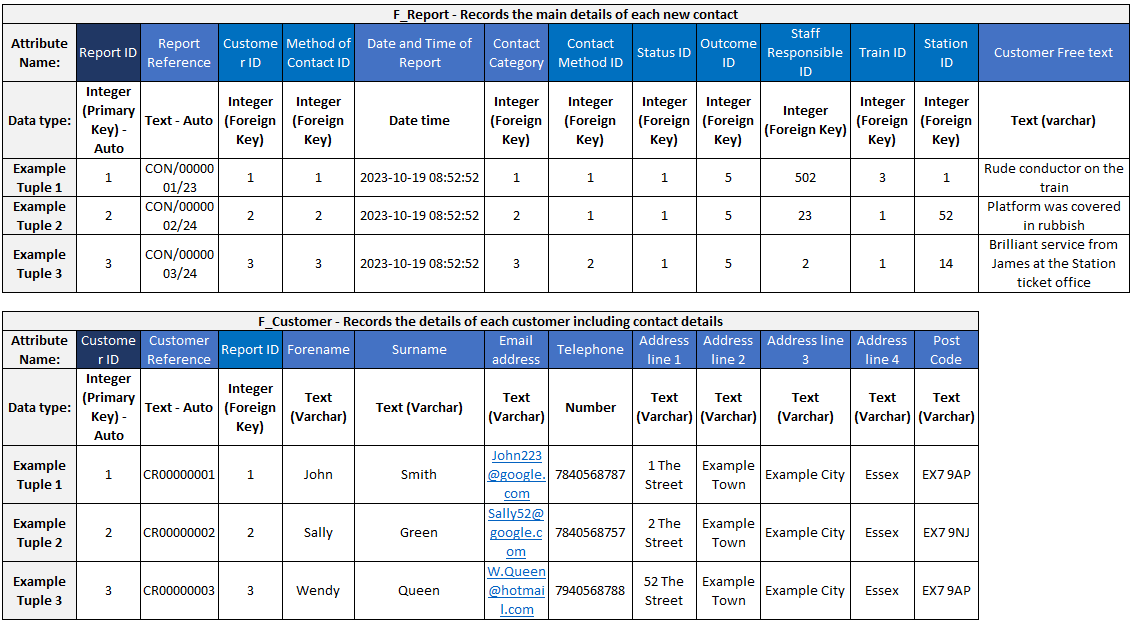
**Introduction**

This report provides an executive summary of the completed design and build of the customer contact management database for a UK rail company.

**Data structure and analysis**

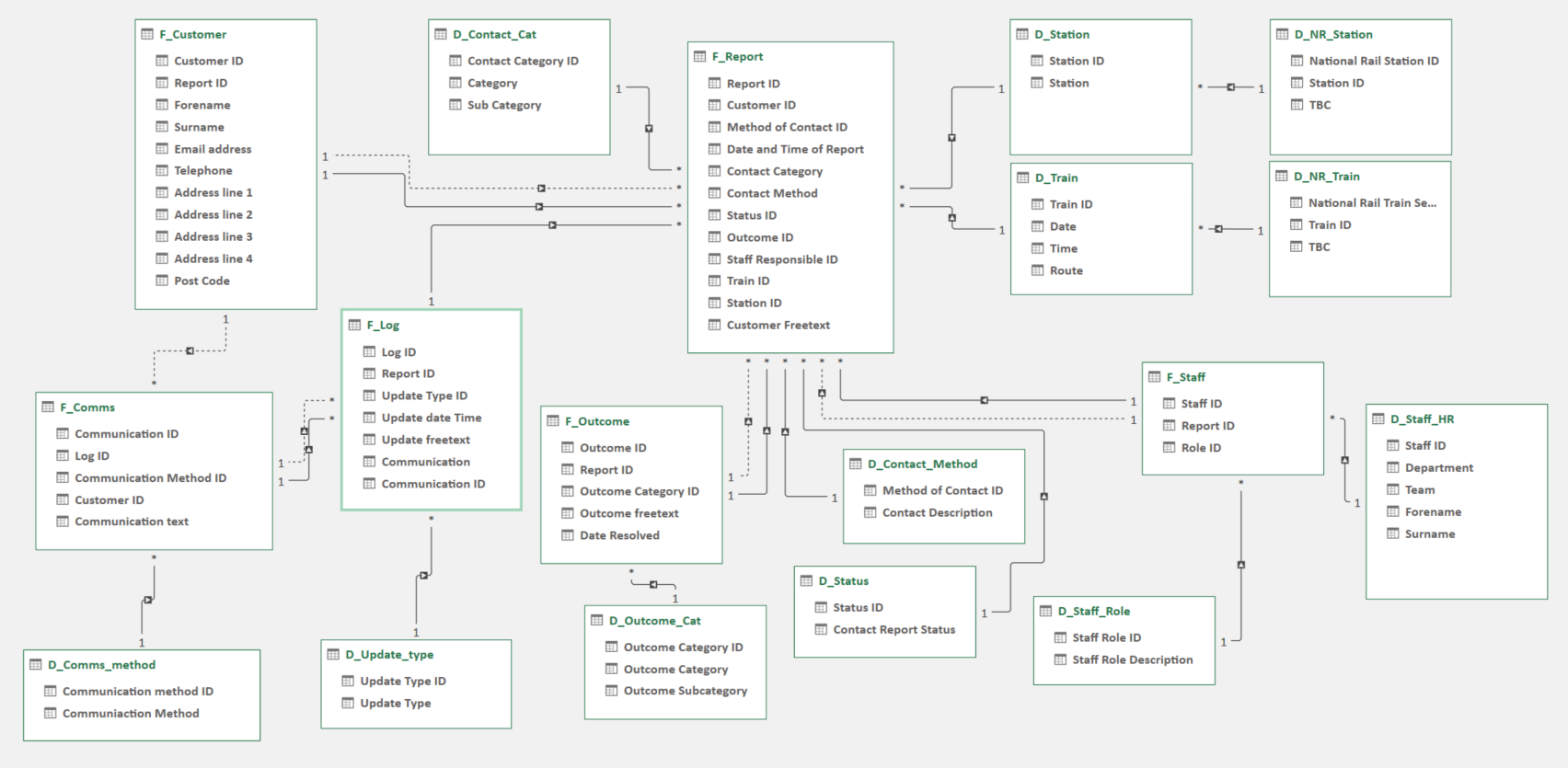
The new customer service platform collects data that is primarily structured data in the form of constrained text values, dates and some longer free text fields, different parts of the process form logical units such as customer information, involved staff and train services. Data processes with links and connections are very suited to being stored a SQL relational database (Kazil & Jarmul, 2016), therefore the system has been built as a Relational Database Management System (RDBMS) using SQL on the Amazon RDS service (AWS Amazon 2023). SQL based RDMS are a highly structured way of storing data in a manner which results in good data integrity and consistency. SQL allows fast access to data; is considered reliable and stable; can handle large volumes of data; and can scale as the size of data or variety of data grows (Welling & Thomson, 2004).

Data in the system has been grouped into relations (tables), with each containing attributes (columns) with an identified data type. Each tuple (row) within a table represents a distinct value that records part of the process and has a unique ID. This is used as an identifier for that row and as a primary key to create links in the database between connected relations (Harrington, 2009). Figure 2 below shows examples for two relations:



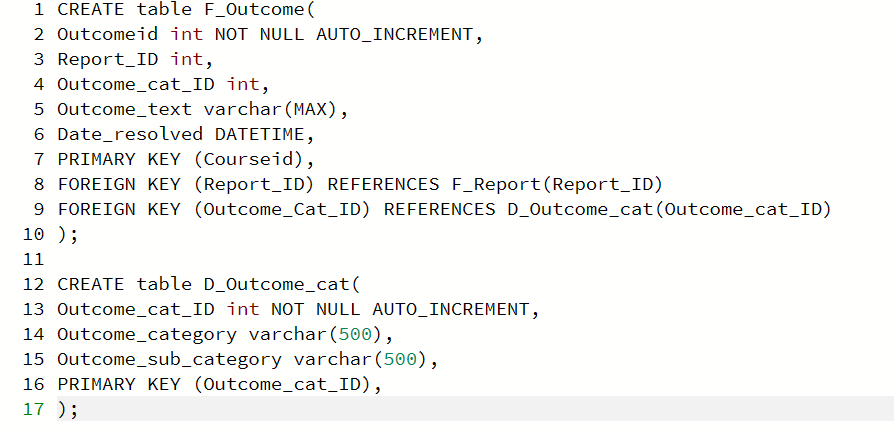
**Figure 2.** Examples of relations

The database design principles of normal forms have been applied at the design and then build stage. Normalisation is intended to organise the system into tables which make logical sense and avoid duplicating data by storing the same data in many tables. (Chapple, 2022) This has the advantage of reducing storage use and improving data integrity. Updates to data need only occur in one table and will apply across the system. If the data was not normalised, an update to a data item such as a phone number in one location would not cascade across the system and member of staff may call the wrong person. Whilst Normalisation is best practice for RDMS, it would not be required for an object orientated solution such as NoSQL (Chapple, 2020) because the data is unstructured. Figure 3 below shows normalisation database design:

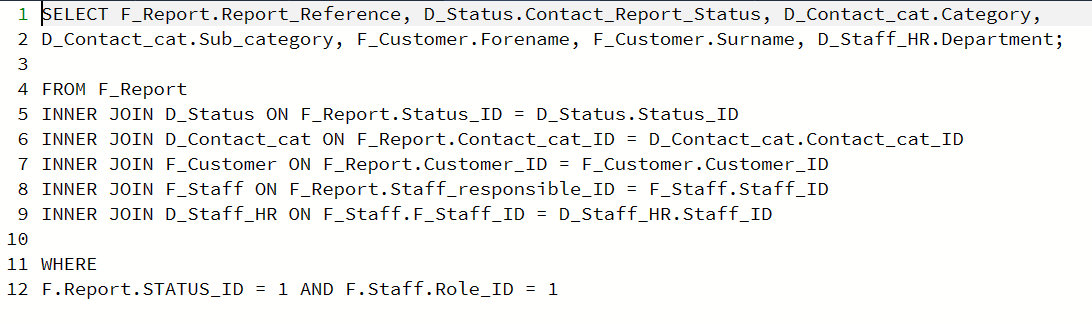


**Figure 3.** Database design

Database Definition Language (DDL) is the element of SQL that allowed the normalised build of the database to be implemented. DDL allowed tables to be created; names and data types of attributes to be defined; column constraint rules to be set; and primary and foreign keys specified. (Harrington, 2009; Sarkar & Roychowdhury, 2019). This coding process is critical for integrity and data quality rules; it ensures the database links together and provides for the basic structure of the system. Figure 4 shows the creation of two tables:



**Figure 4.** Table creation

The new system is intended to enable new data insight about the customer experience in order to drive improvements to the delivery of the service. SQL ensures key data is held in logical structure with data redundancy and data quality issues designed out which aids data analysis by providing more reliable data. SQL uses Data Query Language (DQL) to allow flexible querying of the database to retrieve data to answer business questions. DQL allows bespoke data retrieval by joins between tables, aggregation and selection of specific data items. This allows ad-hoc queries to meet business needs or to create temporary views of the database to allow for regular reporting that can be displayed to users as part of the application (Connolly & Begg, 2014). Figure 5 is SELECT query which provides a list of open contact cases by department with customer details:

**Figure 5.** SELECT query

Python can be used to connect directly to a RDBMS, this provides an option to undertake advanced analysis using a coding language which is highly flexible with libraries designed for Data Science tasks such as data wrangling, statistical analysis and data visualisation (Javin, 2021; Kazil & Jarmul, 2016). The RDMS is compatible with commercial data analysis software, for example AWS offer a suite of data analysis, visualisation and machine learning options (Amazon AWS, 2023). These packages often suitable for less technical users by enabling data analysis without coding, a good option for the organisations in a market where data science skills are highly sought after and not always readily available (World Economic Forum, 2020).

A relational database built with SQL is highly suited to the collection of the type of structured data in a logical process collected as part of this solution. Alternative options using NoSQL would be an option if the collection included unstructured data such as images, or if the volume of data had far greater (Keita, 2022), the priority of collecting data in a structured format made a SQL appropriate. This approach may restrict database scalability in the future to handle big data which is more varied, for example incorporating images or high volume and velocity data from Internet of Things (IoT) devices.

A SQL RDMS provides a range of data analysis options which can scale based on the technical skills and needs of the organisation. Direct SQL queries are highly flexible for retrieving data. Connectivity with Python opens up the possibility of applying advanced analytics techniques. RDMS are also suited to commercial data analysis software which provide capability for staff of varied technical ability. This meets the requirement to enable far more data insight than would have been possible under the legacy platform.

**References:**

AWS. (n.d). What Is AWS Glue?. Amazon Web Services, Inc. Available at: https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html [Accessed 21 October 2023].

AWS. (n.d). Batch-based architecture. In: AWS Well-Architected. Available at: <https://docs.aws.amazon.com/wellarchitected/latest/high-performance-computing-lens/batch-based-architecture.html> [Accessed 21 October 2023].

AWS Amazon (2023). What is a Relational Database? Available from: https://aws.amazon.com/rds/aurora/ [Accessed 20 October 2023]

AWS Amazon (2023). Amazon Aurora Unparalleled high performance and availability at global scale with full MySQL and PostgreSQL compatibility. Available from: https://aws.amazon.com/relational-database/ [Accessed 20 October 2023]

AWS Amazon (2023). Analytics on AWS. Fastest way to get answers from all your data to all your users. Available from: https://aws.amazon.com/big-data/datalakes-and-analytics/ [Accessed 20 October 2023]

AWS (2022) Security best practices in IAM. Available at: <https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#grant-least-privilege>. [Accessed 21 October 2023].

AWS Lambda, Amazon API Gateway, and services from Amazon Web Services. 1st edition. Birmingham, England ;: Packt.

AWS Workshops (n.d). AWS Glue Immersion day. Available at: <https://catalog.us-east-1.prod.workshops.aws/workshops/ee59d21b-4cb8-4b3d-a629-24537cf37bb5/en-US/lab6/etl/advanced-transform> [Accessed 21 October 2023].

Patterson, S. (2019) Learn AWS serverless computing : a beginner’s guide to using

Chapple, M.(2022) Database Normalization Basics, Available from: https://www.lifewire.com/database-normalization-basics-1019735 [Accessed 20 October 2023]

Chapple, M. (2023) Should I Normalize My Database? Available from: https://www.lifewire.com/should-i-normalize-my-database-1019730 [Accessed 20 October 2023]

Connolly, T. M. & Begg, C. E. (2014) Database Systems: A Practical Approach to Design, Implementation and Management. 6th ed. Harlow: Pearson Education Limited.

Harrington, J. L. (2009) Relational database design and implementation clearly explained. 3rd ed. Amsterdam: Morgan Kaufmann/Elsevier.

ICO (n.d) A guide to data security. Available at: https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/security/a-guide-to-data-security/#14. [Accessed 21 October 2023].

ISO 27001 Guide (2016) ISO 27001 Review User Access Rights Requirements. Available at: <https://iso27001guide.com/iso-27001-review-user-access-rights-requirements-iso27001-guide-iso27001-guide.html> [Accessed 21 October 2023].

ICO (2023) A guide to individual rights. Available at: <https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/individual-rights/individual-rights/> [Accessed 21 October 2023].

Javin.P. (2021) Top 5 Reasons to Learn Python for Data Science, Machine Learning, and AI in 2023? Available at: https://medium.com/javarevisited/5-reasons-to-learn-python-for-data-science-16a9d4c44d6d [Accessed 20 October 2023]

Kazil, J. & Jarmul, K. (2016). Data Wrangling with Python. 1st ed. Sebastopol, CA: O’Reilly.

Keita, Z. (2022) NoSQL Databases: What Every Data Scientist Needs to Know. Available at: https://www.datacamp.com/blog/nosql-databases-what-every-data-scientist-needs-to-know [Accessed 21 October 2023]

McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.

Sarkar, T. & Roychowdhury, S. (2019) Data Wrangling with Python. 1st ed. Packt.

World Economic Forum. (2020) The Future of Jobs Report 2020. Available from: https://www.weforum.org/reports/the-future-of-jobs-report-2020 [Accessed 21 October 2023]