

## Evans Cycle Database Proposal

To design our Evans cycle logical database, we would employ both logical and physical data modelling. Our logical data modelling is concerned with the "what", without paying attention to specific functions and capabilities of the database management system (DBMS) that will store the data. Here we documented the comprehensive business information requirements in an accurate and consistent format. Our process of data logical modelling acknowledges that Evans business data is a vital asset we ought to understand and carefully manage. (Sloan , 2004)

In our logical database for Evans cycle, we represented the following business facts in its data model: customers, orders, staff members, stores, order items, categories, products, stocks, and brands. To perform the logical data-modelling design, we began by defining the entities. For example, we defined an entity for all customers because we ought to store information about Evans Cycle's customers. We also define an entity, called orders to store detailed order information. Regarding data entities, Attributes are the specific pieces of information stored within an entity.

Next, we defined the primary keys, which is the unique identifier for the entities. In the case of "customers" and "order" entity, we chose a unique customer id: "customer\_id" and a unique order id "order\_id" as the primary key. We then used the primary key to define separate entities for different types of relationships, which can be one-to-many, many-to-one, one-to-one, or many-to-many. We also decided what values such as domains, null values, and default values are acceptable for the various attributes of the Evans cycle entity. Finally, we normalize the entities to assign the logically related attributes into tables, to avoid update anomalies and minimize redundancy.

In the physical design, we transformed the entities into tables, the instances into rows, and the attributes into columns. After which we implemented the physical design by defining the various objects and enforcing the constraints on the data relationships. (Fong et al., 2008)

Data capture is crucial in database management and employing various methods to ensure the inclusion of data entities in a database. HR records are usually stored in files and payroll systems, captured through dedicated HR processes. Note that discrepancies can result from human errors or incomplete data. Additionally, in-store forms and online sign-ups will serve as the primary sources for customer data. Data capture will occur through these channels, but similar to HR data, maintaining data quality is essential (VisionX, 2023).

Orders and order items are primarily sourced from both 'Point of Sale (POS) systems' and e-commerce 'websites'. This data is captured automatically from the POS system and via APIs from online platforms, enabling integration into our database. Store data originates from staff input and external records, with 'manual data entry' and data file imports used for capturing store details (VisionX, 2023). Suppliers provide data, staff data entry and data feed files will be the primary source of data from Products, Stocks, Brands and categories. It's captured through data feeds for supplier-provided data and manual data entry for product details and updates. With the latter being prone to human error which can lead to incomplete data (VisionX, 2023).

The data collected is from Evans Cycle that sells bikes and was obtained from multiple different source and consists of 9 tables varying from product information, customer details, their stores and employees stored in CSV (Myrick, 2023). This needs to be imported to MS SQL (SQL) server but as it currently stands the data is not readable as it is a CSV file and this is why we can use Python first to clean this data utilising the package Pandas (Agarwal, 2023).

Once imported into Python we convert it to a data frame and will check a few things to clean the data: first we will remove any columns that are not necessary for the table which in this case was already complete; we will then check for any missing values in the data for example in this case some customers didn't provide a phone number and would mark these as NULL; data types in each column need checking and converting them to be readable in SQL in this case we would want to convert the prices of products to a float instead of integers as these need to be precise (Agarwal, 2023).

After cleaning we are ready to import our data to SQL and this will be done by converting our data frame to tuples, as it is ordered and immutable, then export as a txt file and copy this data and insert all of these values.

The cycling industry is experiencing significant growth, and Evans Cycle's success will depend on their ability to adapt to changing demands (Market Size Forecast, 2023). As their customer base and transaction volumes increase over time, the adaptability of MS SQL Server for upgrading server hardware will be a valuable asset (Bernard, 2023).

SQL Server is renowned for its powerful query optimisation capabilities. (Sener, 2023). This ensures that analysts at Evans Cycle can efficiently retrieve data and allocate more time to analytics, facilitating essential business operations and inventory reporting.

The database will accrue a great amount of product listings, customer data, and historic orders which means efficient storage management is crucial. SQL Server provides data compression options that help reduce storage space requirements. This feature allows us to maximise storage efficiency and manage costs (DAssafMSFT, 2023).

SQL Server offers 'backup' and 'restore' functionalities that mitigate data loss risks which are essential for safeguarding the business' data. In addition, its 'data encryption,' 'access' control, and 'authentication' mechanisms align with the GDPR, ensuring the secure processing of sensitive information and transaction records while enabling authorised personnel to access specific database objects (GDPR, undated; Chouhan, 2022).

Third Normal Form (3NF) will be implemented in the database as it helps minimise data redundancy and enhances data integrity (Aslan, 2023). In a cycling store database where information on products, sales, and suppliers are interlinked, applying 3NF ensures that data is organised into separate, related tables, eliminating indirect dependencies (Aslan, 2023). SQL Server's seamless integration of normalised structures complements the logical arrangement of our data (Helencu, 2023).

We've opted for a SQL database due to the 'structured' dataset, which includes product details, customer information, and transactions. SQL excels in handling structured data, ensuring data 'consistency'. NoSQL databases are better suited for 'unstructured' or semi-structured data, which doesn't apply to our structured product, orders, and customer data. Given our specific needs, the 'flexibility' and 'scalability' of NoSQL aren't essential, making SQL the perfect choice for effective data management (Community, 2023).

## References:

Agarwal, M. (2023) Pythonic data cleaning with pandas and NumPy, Real Python. Available at: <https://realpython.com/python-data-cleaning-numpy-pandas/> (Accessed: 15 December 2023).

Aslan, M. (2023) Database normalization, Medium. Available at: <https://medium.com/@murataslan1/database-normalization-e8f1e08c2193> (Accessed: 14 December 2023).

Bernard, A. (2023) Microsoft SQL Server 2022: Here's what you need to know for a successful 2023, The SHI Resource Hub. Available at: <https://blog.shi.com/business-of-it/microsoft-sql-server-2022-heres-what-you-need-to-know-for-a-successful-2023/> (Accessed: 14 December 2023).

Chouhan, P. (2022) SQL Server Vulnerabilities and Assessment, Simple Talk. Available at: <https://www.red-gate.com/simple-talk/databases/sql-server/security/sql-server-vulnerabilities-and-assessment/> (Accessed: 14 December 2023).

Community, S. (2023) Choosing the best database: SQL VS NOSQL, Medium. Available at: <https://medium.com/codex/choosing-the-best-database-sql-vs-nosql-a816a7ec09fc> (Accessed: 14 December 2023).

Fong, J., Shiu, H. and Cheung, D. (2008) 'A relational–XML data warehouse for data aggregation with SQL and XQuery', *Software: Practice and Experience*, 38(11), pp. 1183–1213. doi:10.1002/spe.868.

Global Bicycle Market Size Forecast 2027 (2023) Statista. Available at: <https://www.statista.com/statistics/1356736/bicycle-market-forecast-global/> (Accessed: 14 December 2023).

Guide to the General Data Protection Regulation (GDPR) (no date) [www.ico.org.uk](https://ico.org.uk). Available at: <https://ico.org.uk/media/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr-1-1.pdf> (Accessed: 14 December 2023).

Helencu (2023) Database normalization description - microsoft 365 apps, Database normalization description - Microsoft 365 Apps | Microsoft Learn. Available at: <https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description> (Accessed: 14 December 2023).

Myrick, D. (2023) Bike store relational database: SQL, Kaggle. Available at: <https://www.kaggle.com/datasets/dillonmyrick/bike-store-sample-database> (Accessed: 13 December 2023).

Sener, M.C. (2023) Mastering MSSQL query optimization: Tips and tricks for efficient queries, Medium. Available at: <https://blog.stackademic.com/mastering-mssql-query-optimization-tips-and-tricks-for-efficient-queries-7f55f7007b2d> (Accessed: 14 December 2023).

Sloan , S.G. (2004) 'Page 1 - 6', in The Official Introduction to DB2 for z/OS: Covers DB2 Universal Database for z/OS Version 8. IBM Press, pp. 1–6.

VisionX (2023) What is Data Capture & How Does It Benefit Enterprises, VisionX. Available at: <https://visionx.io/blog/what-is-data-capture/> (Accessed: 16 December 2023).

WilliamDAssafMSFT (2023) Data Compression - SQL Server, SQL Server | Microsoft Learn. Available at: <https://learn.microsoft.com/en-us/sql/relational-databases/data-compression/data-compression?view=sql-server-ver16> (Accessed: 14 December 2023).