1. **Annual Record Count Projections Based on Ridership**

According to the 2023–2024 PTV Annual Report, metropolitan services in Victoria carry the following passenger volumes:

* Buses: 114.9 million
* Trains: 182.5 million
* Trams: 154.8 million

This amounts to approximately 452.2 million passenger trips annually. For estimation purposes, each passenger trip is assumed to correspond to a single **Trip** record in the database (i.e., each tap-on/tap-off event generates one record). Although real-world scenarios often involve multiple taps within a single journey, this assumption serves as a reasonable baseline.

1. **SQL Server Data Type Storage Basics**
2. **INT**: Typically 4 bytes.
3. **DECIMAL(p, s)**: Storage depends on precision and scale. For instance, DECIMAL(10,2) typically requires ~5 bytes.
4. **DATETIME**: 8 bytes.
5. **VARCHAR(n)**: Consumes approximately (number of characters) + 2 bytes of overhead. For instance, storing a 36-character string in VARCHAR(50) would require around 38 bytes.

Each row also incurs **Row Overhead** (metadata such as row headers and column offsets) and **Index Overhead** (additional storage for clustered/non-clustered indexes). These overheads vary considerably based on schema design, indexing strategy, and average fill factors.

1. **Example Calculation: Trip Table**

**3-1. Table Structure**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Notes** |
| trip\_id (PK) | VARCHAR(50) | Primary Key (e.g., GUID or other unique string) |
| card\_id (FK) | VARCHAR(50) | Foreign key referencing MykiCard |
| touch\_on\_time | DATETIME | Datetime of tap-on |
| touch\_on\_scanner\_id | VARCHAR(50) | Scanner used for tap-on |
| touch\_on\_stop\_station\_id | VARCHAR(50) | Stop/station for tap-on |
| touch\_off\_time (nullable) | DATETIME | Datetime of tap-off (null if not tapped off) |
| touch\_off\_scanner\_id (nullable) | VARCHAR(50) | Scanner used for tap-off (if applicable) |
| touch\_off\_stop\_station\_id | VARCHAR(50) | Stop/station for tap-off (if applicable) |
| fare\_charged | DECIMAL(10,2) | Fare amount charged |
| fare\_type | VARCHAR(20) | Type of fare (e.g., full fare, concession) |
| fare\_calc\_method | VARCHAR(20) | Method used to calculate the fare (e.g., 2-hour rule) |

**3-2. Estimated Size per Trip Record**

Assuming an average string length of ~36 characters for VARCHAR(50) columns plus overhead:

1. trip\_id: ~38 bytes
2. card\_id: ~38 bytes
3. touch\_on\_time: 8 bytes
4. touch\_on\_scanner\_id: ~38 bytes
5. touch\_on\_stop\_station\_id: ~38 bytes
6. touch\_off\_time: 8 bytes
7. touch\_off\_scanner\_id: ~38 bytes
8. touch\_off\_stop\_station\_id: ~38 bytes
9. fare\_charged (DECIMAL(10,2)): ~5 bytes
10. fare\_type (VARCHAR(20)): ~12–20 bytes (using 12–15 characters, plus overhead)
11. fare\_calc\_method (VARCHAR(20)): ~12–20 bytes

A preliminary summation might be ~273 bytes before including row overhead and index overhead. Allocating an additional ~10 bytes for row overhead and ~30 bytes for indexes results in approximately 313 bytes per Trip record.

**3-3. Annual Storage for Trip Records**

With an estimated 452.2 million passenger trips per year, and around 313 bytes per record:

452,200,000 records/year × 313 bytes ≈ 141,436,600,000 bytes

≈ 141.44 GB per year

1. **Example Calculation: Transaction Table**

In many systems, each trip may generate zero or one Transaction record (e.g., if a trip is free within a certain time window, no additional transaction is posted). For this example, 80% of trips are assumed to result in a recorded transaction.

**4.1 Table Structure:**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Notes** |
| transaction\_id (PK) | VARCHAR(50) | Primary Key (e.g., GUID) |
| card\_id (FK) | VARCHAR(50) | Foreign key referencing MykiCard |
| trip\_id (nullable FK) | VARCHAR(50) | Foreign key referencing Trip (not always applicable) |
| scanner\_id (FK) | VARCHAR(50) | Scanner or device where the transaction occurred |
| amount | DECIMAL(10,2) | Monetary amount of the transaction |
| type | VARCHAR(20) | Transaction type (e.g., TopUp, FareDeduction, MykiPass) |
| timestamp | DATETIME | When the transaction took place |

**4.2 Estimated Size per Transaction Record**

1. transaction\_id: ~38 bytes
2. card\_id: ~38 bytes
3. trip\_id (nullable): ~38 bytes (assuming full usage for estimation)
4. scanner\_id: ~38 bytes
5. amount (DECIMAL(10,2)): ~5 bytes
6. type (VARCHAR(20)): ~12–20 bytes
7. timestamp (DATETIME): 8 bytes

This initial sum of ~185 bytes can be increased by ~40 bytes to account for row overhead and indexes, resulting in ~225 bytes per transaction record.

**4.3 Annual Storage for Transactions**

If 80% of 452.2 million trips produce a transaction:

452,200,000 × 0.8 ≈ 361,760,000 transaction records/year

Annual storage requirements:

361,760,000 × 225 bytes ≈ 81,396,000,000 bytes ≈ 81.4 GB per year

**5. Summary of Storage Calculation for 2023**

The following table consolidates the rough annual (or total) storage estimates for the primary tables, based on earlier capacity calculations. Detailed assumptions and step-by-step derivations can be referenced in the appendix if needed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | **Estimated Rows** | **Approx. Size per Row** | **Estimated Annual/Total Size** | **Notes** |
| Trip | ~452.2 million per year | ~313 bytes | ~141 GB per year | Driven by annual passenger trips (bus, train, tram). |
| Transaction | ~361.76 million per year | ~225 bytes | ~81 GB per year | Assumes ~80% of trips generate a transaction. |
| CustomerAccount | ~2.4 million total (in 2023 scenario) | ~270 bytes | ~0.65 GB total (not annual) | Grows with population / registration rates. |
| MykiCard | ~2.88 million total | ~170 bytes | ~0.49 GB total (not annual) | Assumes ~1.2 cards per customer account. |
| VehicleStopLog | ~306.6 million per year | ~208 bytes | ~63.8 GB per year | Based on stop logs for buses/trams/trains, assuming ~30 stops per run, multiple runs per day, etc. |
| VehicleRealTimeLog | ~1.96 billion per year | ~172 bytes | ~337 GB per year | Based on frequent GPS position logging (e.g. every 30 seconds, 16 hours/day) for ~2,800 vehicles. |

Based on the estimated record volumes and data types across the major database tables, the projected **annual storage requirement** for 2023 is approximately:

* **Trip**: ~141 GB/year
* **Transaction**: ~81 GB/year
* **VehicleStopLog**: ~63.8 GB/year
* **VehicleRealTimeLog**: ~337 GB/year

This results in a total annual data growth of approximately **623 GB/year**.

In addition, two supporting tables—**CustomerAccount** (~0.65 GB total) and **MykiCard** (~0.49 GB total)—require only modest, non-recurring storage and are not major contributors to ongoing annual growth.

These estimates indicate that the majority of data volume originates from high-frequency logging (**VehicleRealTimeLog**) and large-scale transactional activity (**Trip** and **Transaction** records). While actual storage needs may vary depending on system design (e.g., indexing, compression, retention policies), the system should be provisioned for **multi-terabyte** capacity in the medium to long term.

Appendix:

**1. CustomerAccount**

**1.1 Table Structure**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Notes** |
| customer\_id(PK) | VARCHAR(50) | Unique identifier, e.g., GUID or other string key |
| full\_name | VARCHAR(100) | Full name of the account holder |
| email | VARCHAR(50) | Email address |
| phone\_num | VARCHAR(20) | Phone number |
| address | VARCHAR(100) | Customer’s address |
| dob | DATE | Date of birth |
| register\_date | DATETIME | Date/time the account was created |
| account\_status | VARCHAR(20) | Status of the account (active/inactive, etc.) |
| password\_hash | VARCHAR(256) | Storage for hashed password |

### 1.2 Estimated Size per Record

* customer\_id (VARCHAR(50)): ~38 bytes
* full\_name (VARCHAR(100)): assume ~20–30 characters average ⇒ ~22–32 bytes
* email (VARCHAR(50)): assume ~25 characters average ⇒ ~27 bytes
* phone\_num (VARCHAR(20)): assume ~10–15 characters ⇒ ~12–17 bytes
* address (VARCHAR(100)): assume ~40 characters ⇒ ~42 bytes
* dob (DATE): 3 bytes
* register\_date (DATETIME): 8 bytes
* account\_status (VARCHAR(20)): assume ~10 characters ⇒ ~12 bytes
* password\_hash (VARCHAR(256)): can vary widely. For a typical hashed password (e.g., 64 hex characters), estimate ~66 bytes

A rough summation might be around 230 bytes of field data.  
Adding row overhead (~10 bytes) and index overhead (~30 bytes), each record is approximately:

230 + 10 + 30 = ~270 bytes

### 1.3 Total Storage Estimate

Using population data from the Australian Bureau of Statistics (2023), Victoria has roughly 6.8 million residents, with 70% (≈4.76 million) in Greater Melbourne. If half of those residents eventually register for a CustomerAccount, that yields about 2.4 million accounts.

2,400,000 accounts × 270 bytes ≈ 648,000,000 bytes

≈ 0.65 GB

By 2051, the projected population is 10.3 million, which, at the same 70% in Greater Melbourne and a similar registration rate, might bring the total to around 3.6 million accounts—still within the low gigabyte range for storage.

## 2. MykiCard

### 2.1 Table Structure

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Notes** |
| card\_id (PK) | VARCHAR(50) | Unique identifier (could be a GUID) |
| balance | DECIMAL(10,2) | Current monetary balance on the card |
| customer\_id(FK, nullable) | VARCHAR(50) | Links to CustomerAccount if not anonymous |
| card\_type | VARCHAR(20) | E.g., Full-fare, Concession, Senior, etc. |
| myki\_pass | VARCHAR(50) | Details of any pass linked to the card |

### 2.2 Estimated Size per Record

* card\_id (VARCHAR(50)): ~38 bytes
* balance (DECIMAL(10,2)): ~5 bytes
* customer\_id (VARCHAR(50)): ~38 bytes (nullable, but assume worst-case usage)
* card\_type (VARCHAR(20)): assume ~10–15 characters ⇒ ~12–20 bytes
* myki\_pass (VARCHAR(50)): assume ~25 characters average ⇒ ~27 bytes

Summing to roughly 120–130 bytes before overhead. Adding ~10 bytes row overhead and ~30 bytes index overhead:

~130 + 10 + 30 = ~170 bytes per record

### 2.3 Total Storage Estimate

If each CustomerAccount has on average 1.2 MykiCards (including anonymous cards in the overall system), the total number of cards is roughly:

2,400,000 accounts × 1.2 = 2,880,000 cards

Hence,

2,880,000 cards × 170 bytes ≈ 489,600,000 bytes

≈ 0.49 GB

Even if the number of cards grows proportionally with the population, the storage demand remains in the sub-gigabyte to low-gigabyte range, which is significantly smaller compared to tables like Trip and Transaction.

## 3. VehicleStopLog

### 3.1 Table Structure

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Notes** |
| stop\_log\_id (PK) | VARCHAR(50) | Unique identifier for the stop log event |
| vehicle\_id (FK) | VARCHAR(50) | Links to the Vehicle |
| run\_id (FK) | VARCHAR(50) | Links to a particular vehicle run (VehicleRun) |
| stop\_station\_id(FK) | VARCHAR(50) | Which station or stop the vehicle arrived at |
| actual\_arrival\_time | DATETIME | Actual time of arrival |
| actual\_departure\_time | DATETIME | Actual time of departure |

### 3.2 Estimated Size per Record

* stop\_log\_id: ~38 bytes
* vehicle\_id: ~38 bytes
* run\_id: ~38 bytes
* stop\_station\_id: ~38 bytes
* actual\_arrival\_time (DATETIME): 8 bytes
* actual\_departure\_time (DATETIME): 8 bytes

Summation of field data is around 168 bytes. Adding ~10 bytes row overhead plus ~30 bytes index overhead yields approximately:

168 + 10 + 30 = ~208 bytes per record

### 3.3 Total Storage Estimate

An approximate strategy for calculating annual logs involves estimating the number of runs and stops:

1. Suppose there are about 2,800 total vehicles (including buses, trams, trains).
2. Assume each vehicle performs 10 runs per day, leading to 10 × 365 = 3,650 runs per vehicle per year.
3. Each run might have an average of 30 stops.

Hence, for each vehicle:

3,650 runs/year × 30 stops/run = 109,500 stop logs/year/vehicle

Across 2,800 vehicles:

109,500 × 2,800 ≈ 306,600,000 stop logs/year

Multiplying by the record size:

306,600,000 × 208 bytes ≈ 63,772,800,000 bytes

≈ 63.8 GB per year

## 4. VehicleRealTimeLog

### 4.1 Table Structure

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Notes** |
| live\_log\_id (PK) | VARCHAR(50) | Unique identifier for each real-time GPS log event |
| vehicle\_id (FK) | VARCHAR(50) | Links to the Vehicle |
| stop\_station\_id(FK, nullable) | VARCHAR(50) | Station or stop, if the vehicle is currently at one (optional) |
| vehicle\_lat | DECIMAL(9,6) | Current latitude of the vehicle |
| vehicle\_long | DECIMAL(9,6) | Current longitude of the vehicle |
| timestamp | DATETIME | Timestamp of the real-time log |

### 4.2 Estimated Size per Record

* live\_log\_id: ~38 bytes
* vehicle\_id: ~38 bytes
* stop\_station\_id (nullable): ~38 bytes (worst-case assumption)
* vehicle\_lat (DECIMAL(9,6)): ~5 bytes
* vehicle\_long (DECIMAL(9,6)): ~5 bytes
* timestamp (DATETIME): 8 bytes

This sums to about 132 bytes. Adding ~10 bytes row overhead plus ~30 bytes for indexing:

132 + 10 + 30 = ~172 bytes per record

### 4.3 Total Storage Estimate

One possible assumption is that each vehicle reports its position every 30 seconds while in service, for 16 hours per day:

1. 16 hours = 57,600 seconds, at 30-second intervals = 1,920 logs per vehicle per day.
2. For 2,800 vehicles, that totals:

1,920 logs/day/vehicle × 2,800 vehicles = 5,376,000 logs/day

1. Over one year (~365 days):

5,376,000 × 365 ≈ 1.96 billion logs/year

Multiplying by ~172 bytes per log:

1,960,000,000 × 172 bytes ≈ 337,120,000,000 bytes

≈ 337 GB per year

Again, these numbers can fluctuate considerably depending on the actual number of vehicles, their operational hours, and the logging frequency.