**1. Get your data into Power BI**

1. **Open Power BI Desktop**
2. **Get Data → Text/CSV** → load both CSVs:

A screenshot of a computer

AI-generated content may be incorrect.

* + **CardAccessGroupAssignment.csv**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* + **CardTransaction.csv**

A screenshot of a computer

AI-generated content may be incorrect.

1. In the **Navigator**, select each file and click **Transform Data** to open Power Query.

A screenshot of a computer

AI-generated content may be incorrect.

**2. Clean & shape your tables in Power Query**

**A. CardTransaction table**

1. **Parse the date/time**
   * Ensure **EntranceTime** and **ExitTime** are DateTime types.

A screenshot of a computer

AI-generated content may be incorrect.

* As it is still in ‘Whole Number’, we need to change it to Date/Time.
* To do that Click on the “ABC” and select “Date/Time”.

A screenshot of a computer

AI-generated content may be incorrect.

* Do the same for the ‘ExitTime’ Column as well.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Fill missing times**
   * For rows with no EntranceTime:
     + Add a custom column with column name: ActualEntry and insert the code to ensure if there’s no EntranceTime, treat entry as midnight of the exit date.

ActualEntry = if [EntranceTime] = null

then DateTime.From(Date.From([ExitTime]))

else [EntranceTime]

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* + For rows with no ExitTime:
    - Add another custom column with the column name: ActualExit and insert the code to ensure exit at 23:59:59 if ExitTime is missing.

ActualExit = if [ExitTime] = null

then DateTime.From(Date.From([EntranceTime]) & #time(23,59,59))

else [ExitTime]

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Calculate duration**
   * Add a custom column with the column name: DurationMinutes and insert the code to ensure

DurationMinutes = Duration.TotalMinutes([ActualExit] - [ActualEntry])

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. To count how many cars are “in” the lot at any time, we turn each entry & exit into +1/–1 events.

**Duplicate** your CardTransaction query twice (right-click → Duplicate).

A screenshot of a computer

AI-generated content may be incorrect.

In the first copy:

* + Remove all columns except **ActualEntry** and **LotNumber**.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* + Rename **ActualEntry** → **EventTime** by right Clicking on the ActualEntry Column and select rename.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a calendar

AI-generated content may be incorrect.

* + **Add Column → Custom Column**, call it **Delta**, with value 1.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a calendar

AI-generated content may be incorrect.

Make sure you create a second copy named “Events Copy” following the same steps

* + Keep **ActualExit** and **LotNumber** only.

A screenshot of a computer

AI-generated content may be incorrect.

* + Rename **ActualExit** → **EventTime**.

A screenshot of a calendar

AI-generated content may be incorrect.

* + **Add Column → Custom Column**, call it **Delta**, with value -1.

A screenshot of a computer

AI-generated content may be incorrect.

**Append Queries**:

* + On the left, right-click the first **Events** Table→ **Append Queries → Append as New**.

A screenshot of a computer

AI-generated content may be incorrect.

* + Choose the second Events copy to append.

A screenshot of a computer

AI-generated content may be incorrect.

* + You now have one **Events** table with every +1 (entry) and –1 (exit).

**B. CardAccessGroupAssignment table**

1. Ensure **CardNumber** and **AccessGroup** are the correct data types (text or whole number). Made sure the **CardNumber** column is text (or whole number) and the **AccessGroup** column is text (or whole number), whichever matches how you’ll join to the transactions. If Power BI treats one as text and the other as number, the relationship won’t work. Consistent types let you link every card to its group(s).

A screenshot of a computer

AI-generated content may be incorrect.

1. If a card has multiple groups in one row, **split** that column into multiple rows (Home → Split Column → By Delimiter → “;”).

A screenshot of a computer

AI-generated content may be incorrect.

This “normalizes” the table into a true one-row-per-card-per-group format. It makes it easy to answer questions like “Which cards in group 205 hit Lot A?” or “How many group 101 users contributed to the peak on April 15?” without awkward text-parsing or aggregated lists.

A screenshot of a computer

AI-generated content may be incorrect.

Finally click **Close & Apply**.

**3. Build your data model**

1. **Calendar table**

 **Time intelligence**: Allows you to slice and dice metrics by day, week, month, quarter, or year.

 **Consistency**: Even if some dates have no transactions, the calendar still “knows” every date exists.

 **DAX features**: Many built-in time functions (e.g., TOTALYTD, SAMEPERIODLASTYEAR) require a dedicated date table.

* + Home → Modeling → New table, use DAX:

A screenshot of a computer

AI-generated content may be incorrect.

Calendar =

ADDCOLUMNS(

CALENDAR(DATE(2021,1,1), DATE(2025,12,31)),

"Year", YEAR([Date]),

"Month", FORMAT([Date],"MMM YYYY"),

"DayOfWeek", FORMAT([Date],"dddd")

)

* + Press Enter. Your new “Calendar” table will show up in the Fields pane.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Relationships**

* In the left sidebar, click the **Model** icon (it looks like three boxes connected).
* Drag **Events[EventDate]** (EventTime converted to Date) onto **Calendar[Date]**.
* Drag **CardTransaction[CardNumber]** onto **CardAccessGroupAssignment[CardNumber]**.
* Ensure each relationship is **one-to-many** and **single direction** (from Calendar → Events, from Card → Assignments).
  + **Events[EventTime]** → **Calendar[Date]** (Date part only)
  + **CardTransaction[CardNumber]** → **CardAccessGroupAssignment[CardNumber]**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Creating New dimension to link card number as it has duplicates

A screenshot of a computer

AI-generated content may be incorrect.

**Create a “Card” table**

* In Power BI Desktop, go to **Modeling → New Table** and enter:

Card = DISTINCT( CardTransaction[CardNumber] )

**Link Transaction → Card**

A screenshot of a computer

AI-generated content may be incorrect.

* In Model view, drag **CardTransaction[CardNumber]** onto **Card[CardNumber]**.

In the dialog:

* + Cardinality: **Many (Transaction) to one (Card)**
  + Cross-filter: **Single** (Card filters Transactions)
  + Active: ✅

A screenshot of a computer

AI-generated content may be incorrect.

**Link Assignment → Card**

* Drag **CardAccessGroupAssignment[CardNumber]** onto **Card[CardNumber]**.
* In the dialog:
  + Cardinality: **Many (Assignment) to one (Card)**
  + Cross-filter: **Single** (Card filters Assignments)
  + Active: ✅

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**4. Create your core measures in DAX**

**A. Running total of active cards**

* In Power BI Desktop, go to the Modeling tab → New Measure.
* Name it and paste in this DAX:

**A screenshot of a computer

AI-generated content may be incorrect.**

CumulativeActive =

VAR SortedEvents =

SUMMARIZE(

'Events',

'Events'[EventTime],

"NetDelta", SUM('Events'[Delta])

)

RETURN

CALCULATE(

SUMX(SortedEvents, [NetDelta]),

FILTER(

ALLSELECTED('Events'),

'Events'[EventTime] <= MAX('Events'[EventTime])

)

)

**B. Peak usage by period**

Why we created the EventDate column

1. Isolate the date
   * Our original EventTime is a full DateTime (e.g. “2022-03-15 08:37:21”).
   * To group or filter by calendar date only—ignoring time—we need a pure Date field.
2. Enable correct filtering
   * Our DailyPeak measure will look at all entry/exit events for a single date.
   * If we tried to compare full DateTime values to a date (from the Calendar table), it wouldn’t match.
3. Support the Calendar relationship
   * We linked Calendar[Date] ↔ Events[EventDate] so any slicer or axis on Calendar flows down to Events.
   * Without EventDate, that relationship would have to use the full DateTime column, which breaks standard time-intelligence patterns.

**A screenshot of a computer

AI-generated content may be incorrect.**

**How it works**

* CurrDate = MAX(Calendar[Date]) gives you the one date in the current row or filter context.
* ALL(Events) clears existing filters so we can reapply exactly that one date.
* ADDCOLUMNS(..., "\_\_Count", [CumulativeActive]) computes your running-total for every timestamp.
* FILTER(..., Events[EventDate] = CurrDate) keeps only that day’s events.
* MAXX(..., [\_\_Count]) finds the highest count on that day.

**Creating a snapshot table of “Active cars at each timestamp”**

**Dax:**

CopyEdit

CumulativeHistory =

ADDCOLUMNS(

Events,

"ActiveCount", [CumulativeActive]

)

**Daily Peak**

DailyPeak =

VAR CurrDate = MAX( Calendar[Date] )

RETURN

MAXX(

FILTER(

ADDCOLUMNS(

ALL( Events ),

"\_\_Count", [CumulativeActive]

),

Events[EventDate] = CurrDate

),

[\_\_Count]

] )

A screenshot of a computer

AI-generated content may be incorrect.

* **Weekly Peak**

**Dax:**

WeeklyPeak =

VAR CurrDate = MAX( Calendar[Date] )

VAR CurrYear = YEAR( CurrDate )

VAR CurrWeek = WEEKNUM( CurrDate, 2 ) // ISO week (Mon-Sun); use 1 for Sun-Sat

RETURN

MAXX(

FILTER(

ADDCOLUMNS(

ALL( Events ),

"\_\_Count", [CumulativeActive],

"\_\_EvtYear", YEAR( Events[EventDate] ),

"\_\_EvtWeek", WEEKNUM( Events[EventDate], 2 )

),

[\_\_EvtYear] = CurrYear

&& [\_\_EvtWeek] = CurrWeek

),

[\_\_Count]

)

A screenshot of a computer

AI-generated content may be incorrect.

* **Monthly Peak**

**dax**

CopyEdit

MonthlyPeak =

MAXX(

VALUES(Calendar[Month]),

CALCULATE(MAX([CumulativeActive]))

)

A screenshot of a computer

AI-generated content may be incorrect.

**5. Build your report pages**

**Page 1: Overview & Trends**

* **Line chart** of **CumulativeActive** over time (use the Calendar[Date] axis).
  + Enable the built-in **Analytics → Forecast** (set 8 months for May–Dec 2025).
* **Card visuals** showing:
  + **Max Daily Peak**
  + **Max Monthly Peak**

**Page 2: Lot Drill-Down**

* **Slicer** for **LotNumber** (from CardTransaction).
* **Clustered column chart**:
  + Axis → Calendar[Month] (hierarchy)
  + Value → **MonthlyPeak**
  + Enable **Drill down** so clicking a month shows daily peaks.
* **Table** or **Matrix**:
  + Rows → AccessGroup
  + Values → CALCULATE([DailyPeak], CardAccessGroupAssignment[AccessGroup]=EARLIER(...))
  + Use this to see which groups contribute most.

**Page 3: Detail Drill-Through**

1. Right-click on any lot or month → **Drill through → Detailed**
2. On the Detail page, place:
   * **Date slicer** (drill-through filters)
   * **Line chart** of **CumulativeActive** by hour of day
   * **Table** of top 10 cards by total duration in that period

**6. Enhance interactivity**

* **Sync slicers** across pages (View → Sync slicers).
* **Bookmarks & Buttons** to toggle between Daily/Weekly/Monthly views.
* **Tooltips**: create a tooltip page showing “Peak time of day” for a hover over any point.

**7. Add context layers**

* **Import a table** of academic calendar events; join on date and overlay as vertical reference lines on your line charts.
* (Optional) **Web-fetch** weather data and relate by date to see if storms correlate with parking dips.

**8. Publish & share**

1. **Save** your .pbix file.
2. **Publish** to Power BI Service.
3. In the service, set up **scheduled refresh** on your CSV source.
4. Share the dashboard URL with administrators, granting them **Viewer** or **Contributor** access as needed.