Power BI Cheat Sheet

**1. Aggregation Functions**

| **Function** | **Syntax** | **Description** |
| --- | --- | --- |
| SUM | SUM(<Column>) | Sum of all values in a column. |
| SUMX | SUMX(<Table>, <Expression>) | Sum of an expression evaluated row-by-row. |
| AVERAGE | AVERAGE(<Column>) | Arithmetic mean. |
| AVERAGEX | AVERAGEX(<Table>, <Expression>) | Mean of an expression evaluated per row. |
| MIN | MIN(<Column>) | Smallest value. |
| MAX | MAX(<Column>) | Largest value. |
| COUNT | COUNT(<Column>) | Count of non-blank values. |
| COUNTA | COUNTA(<Column>) | Count of all non-empty cells. |
| COUNTROWS | COUNTROWS(<Table>) | Number of rows in a table. |
| DISTINCTCOUNT | DISTINCTCOUNT(<Column>) | Count of distinct values. |

**2. Logical Functions**

| **Function** | **Syntax** | **Description** |
| --- | --- | --- |
| IF | IF(<Condition>, <TrueResult>, <FalseResult>) | Returns one of two values by evaluating a test. |
| SWITCH | SWITCH(<Expression>, <Value1>, <Result1>, …, <Else>) | Multiple-branch logic. |
| AND | AND(<Cond1>, <Cond2>, …) | Returns TRUE if all arguments are true. |
| OR | OR(<Cond1>, <Cond2>, …) | Returns TRUE if any argument is true. |
| NOT | NOT(<Condition>) | Logical negation. |

**3. Filter and Row Context Functions**

| **Function** | **Syntax** | **Description** |
| --- | --- | --- |
| FILTER | FILTER(<Table>, <Condition>) | Returns a table filtered by condition. |
| ALL | ALL(<TableOrColumn>) | Ignores filters (i.e., returns all rows/values). |
| ALLEXCEPT | ALLEXCEPT(<Table>, <Column1>, …) | Ignores all filters except on specified columns. |
| ALLSELECTED | ALLSELECTED(<TableOrColumn>) | Removes filters but preserves filters from outside visuals. |
| VALUES | VALUES(<Column>) | Returns distinct values in column, respects filters. |
| RELATED | RELATED(<Column>) | Returns a related value from another table. |
| RELATEDTABLE | RELATEDTABLE(<Table>) | Returns a table of related rows. |
| EARLIER | EARLIER(<Column>, <N>) | Allows nested row-context in calculated columns. |

**4. Time Intelligence Functions**

**Note:** Requires a proper Date dimension table marked as “Date” in your model.

| **Function** | **Syntax** | **Description** |
| --- | --- | --- |
| DATESYTD | DATESYTD(<DateColumn>[, <YearEndDate>]) | Dates from start of year to current context date. |
| SAMEPERIODLASTYEAR | SAMEPERIODLASTYEAR(<DateColumn>) | Same period in previous year. |
| PARALLELPERIOD | PARALLELPERIOD(<DateColumn>, <NumberOfIntervals>, <Interval>) | Shift by intervals (Year, Quarter, Month). |
| DATEADD | DATEADD(<DateColumn>, <Number>, <Interval>) | Shift dates by given interval. |
| TOTALYTD | TOTALYTD(<Expression>, <DateColumn>[, <YearEndDate>]) | YTD total of an expression. |
| DATESBETWEEN | DATESBETWEEN(<DateColumn>, <StartDate>, <EndDate>) | Returns dates between two dates. |
| FIRSTDATE / LASTDATE | FIRSTDATE(<DateColumn>) LASTDATE(<DateColumn>) | First/last date in current context. |

**5. Text Functions**

| **Function** | **Syntax** | **Description** |
| --- | --- | --- |
| CONCATENATE | CONCATENATE(<Text1>, <Text2>) | Join two text values. |
| COMBINEVALUES | COMBINEVALUES(<Delimiter>, <Text1>, <Text2>, …) | Fast multi-text join. |
| UNICHAR | UNICHAR(<Number>) | Returns the unicode character. |
| UNICHAR | UNICODE(<Text>) | Returns numeric Unicode code for first character. |
| LEFT / RIGHT | LEFT(<Text>, <NumChars>) / RIGHT(<Text>, <NumChars>) | Extract characters. |
| MID | MID(<Text>, <Start>, <Length>) | Substring from text. |
| LEN | LEN(<Text>) | Text length. |
| TRIM | TRIM(<Text>) | Remove leading/trailing spaces. |
| UPPER / LOWER | UPPER(<Text>) / LOWER(<Text>) | Change case. |

**6. Statistical Functions**

| **Function** | **Syntax** | **Description** |
| --- | --- | --- |
| MEDIAN | MEDIAN(<Column>) | Median of values. |
| PERCENTILE.EXC | PERCENTILE.EXC(<Column>, <Percent>) | Percentile excluding endpoints. |
| PERCENTILE.INC | PERCENTILE.INC(<Column>, <Percent>) | Percentile including endpoints. |
| RANKX | RANKX(<Table>, <Expression>[, <Value>[, <Order>]]) | Rank values in table. |
| TOPN | TOPN(<N>, <Table>, <OrderBy>[, <Order>]) | Returns top N rows. |

**7. Iterator Functions**

| **Function** | **Syntax** | **Description** |
| --- | --- | --- |
| SUMX | SUMX(<Table>, <Expression>) | Sum over table. |
| AVERAGEX | AVERAGEX(<Table>, <Expression>) | Average over table. |
| MINX / MAXX | MINX(<Table>, <Expression>) | Min/Max over table. |
| COUNTX | COUNTX(<Table>, <Expression>) | Count non-blank results of expression. |
| FILTER | FILTER(<Table>, <Condition>) | Table filtered by condition (iterator). |

**8. Information & Parent-Child**

| **Function** | **Syntax** | **Description** |
| --- | --- | --- |
| ISBLANK | ISBLANK(<Value>) | TRUE if blank. |
| ISERROR | ISERROR(<Value>) | TRUE if error. |
| HASONEVALUE | HASONEVALUE(<Column>) | TRUE if exactly one distinct value in context. |
| PATH | PATH(<ParentColumn>, <ChildColumn>) | Create parent-child path string. |
| PATHITEM | PATHITEM(<Path>, <Position>, <Delimiter>) | Extract nth node from path. |

**9. Calendar & Date Creation**

| **Function** | **Syntax** | **Description** |
| --- | --- | --- |
| CALENDAR | CALENDAR(<StartDate>, <EndDate>) | Returns a table with one date column. |
| CALENDARAUTO | CALENDARAUTO([<FiscalYearEndMonth>]) | Auto date table based on model data. |
| TODAY | TODAY() | Current date. |
| NOW | NOW() | Current date & time. |
| YEAR/MONTH/DAY | YEAR(<Date>), MONTH(<Date>), DAY(<Date>) | Extract parts of a date. |

**10. Common Patterns**

-- 1. Simple Measure

Total Sales = SUM(Sales[Amount])

-- 2. Conditional Measure

High Value Sales = CALCULATE(

SUM(Sales[Amount]),

Sales[Amount] > 1000

)

-- 3. Year-to-Date

Sales YTD = TOTALYTD(

SUM(Sales[Amount]),

'Date'[Date]

)

-- 4. % of Total

% of Total Sales = DIVIDE(

SUM(Sales[Amount]),

CALCULATE(SUM(Sales[Amount]), ALL(Sales)),

0

)

-- 5. Running Total

Running Sales = CALCULATE(

SUM(Sales[Amount]),

FILTER(

ALLSELECTED('Date'),

'Date'[Date] <= MAX('Date'[Date])

)

)

-- 6. Rank by Sales

Sales Rank = RANKX(

ALL(Customer),

[Total Sales],

,

DESC,

Skip

)

**Usage Tips**

* **Keep your Date table** and mark it as a Date table in the model.
* **Use CALCULATE** to change filter context.
* **Use DIVIDE** (instead of /) to avoid divide-by-zero errors.
* **Leverage IntelliSense** in Power BI Desktop for syntax hints.
* **Group similar measures** in folders for easier navigation.

Other Technologies

**✅ Final Conclusion for Your Prompt:**

Adi, you're already on the right track by learning **DSA** and planning to dive into the **MERN stack**. To truly **impress interviewers** and aim for a **higher-paying role**, consider adding 1–2 of these high-impact skills alongside your core learning:

* 🏗 **System Design** – for cracking senior-level rounds
* ☁ **Cloud (AWS/GCP/Azure)** – for real-world deployment expertise
* 🐳 **DevOps (Docker, CI/CD)** – for smoother and professional workflows
* 🧠 **Advanced Data Structures, Design Patterns, and Security** – for depth in software thinking
* 📈 **Soft Skills, Project Building, & Clean Code** – for standing out as a complete developer

By learning **Power BI** alongside for 1 hour/day, you also open doors into **data-driven roles** and dashboard-based projects—giving your resume a unique edge.

💡 Focus on **project-based learning** and build a **portfolio** that shows your skills in action.

Would you like a full **roadmap** that includes DSA + MERN + Cloud + System Design in one unified plan?