```python
import pandas as pd

# Replace 'your_file.xlsx' with the path to your Excel file
data = pd.read_excel('britishairway.xlsx')
data.head(3)
```

	FLIGHT_DATE	FLI...	TI...	A...	F.	DEPARTURE_STATI...	ARRIVAL_STATI...
0	2025-09-02T00:00:00.000	14:19:00	Afternoon	BA	BA5211	LHR	LAX
1	2025-06-10T00:00:00.000	06:42:00	Morning	BA	BA7282	LHR	LAX
2	2025-10-27T00:00:00.000	15:33:00	Afternoon	BA	BA1896	LHR	FRA

Rows: 3 Expand

```python
# Calculate total passengers
data["TOTAL_PAX"] = data["FIRST_CLASS_SEATS"] + data["BUSINESS_CLASS_SEATS"] + data["ECONOMY_SEATS"]

# Assign lounge eligibility based on BA policy
# Tier 1: Concorde Room (First Class)
data["TIER1_ELIGIBLE_PAX"] = data["FIRST_CLASS_SEATS"]

# Tier 2: First Lounge (Gold members, status-based)
GOLD_RATE = 0.05   # assume 5% of all passengers are Gold
data["TIER2_ELIGIBLE_PAX"] = (data["TOTAL_PAX"] * GOLD_RATE).round()

# Tier 3: Club Lounge (Business seats + Silver members from Economy)
SILVER_RATE = 0.15  # assume 15% of economy passengers are Silver
data["TIER3_ELIGIBLE_PAX"] = data["BUSINESS_CLASS_SEATS"] + (data["ECONOMY_SEATS"] * SILVER_RATE).round()

# Summarize by category (time of day, haul, region)
summary = data.groupby(["TIME_OF_DAY", "HAUL", "ARRIVAL_REGION"])[
    ["TIER1_ELIGIBLE_PAX", "TIER2_ELIGIBLE_PAX", "TIER3_ELIGIBLE_PAX"]
].sum().reset_index()

display(summary)
```

index	TIME_OF_DAY	HAUL	ARRIVAL_REGION	TIER1_ELIGIBLE_PAX	TIER2_E
0	Afternoon	LONG	Asia	582	
1	Afternoon	LONG	Middle East	576	
2	Afternoon	LONG	North America	2284	
3	Afternoon	SHORT	Europe	0	
4	Evening	LONG	Asia	790	
5	Evening	LONG	Middle East	860	
6	Evening	LONG	North America	3068	
7	Evening	SHORT	Europe	0	
8	Lunchtime	LONG	Asia	284	
9	Lunchtime	LONG	Middle East	332	
10	Lunchtime	LONG	North America	1118	
11	Lunchtime	SHORT	Europe	0	
12	Morning	LONG	Asia	920	
13	Morning	LONG	Middle East	842	
14	Morning	LONG	North America	3816	
15	Morning	SHORT	Europe	0	

Rows: 16 Expand

```python
# Step 3: Aggregate by grouping
summary = data.groupby(["TIME_OF_DAY", "HAUL", "ARRIVAL_REGION"])[
    ["TIER1_ELIGIBLE_PAX", "TIER2_ELIGIBLE_PAX", "TIER3_ELIGIBLE_PAX"]
].sum().reset_index()

# Step 4: Calculate percentages
summary["TOTAL"] = summary["TIER1_ELIGIBLE_PAX"] + summary["TIER2_ELIGIBLE_PAX"] + summary["TIER3_ELIGIBLE_PAX"]
summary["Tier1_%"] = (summary["TIER1_ELIGIBLE_PAX"] / summary["TOTAL"] * 100).round(0).astype(str) + "%"
summary["Tier2_%"] = (summary["TIER2_ELIGIBLE_PAX"] / summary["TOTAL"] * 100).round(0).astype(str) + "%"
summary["Tier3_%"] = (summary["TIER3_ELIGIBLE_PAX"] / summary["TOTAL"] * 100).round(0).astype(str) + "%"

# Step 5: Add example destinations (first 3 unique arrival stations per group)
examples = data.groupby(["TIME_OF_DAY", "HAUL", "ARRIVAL_REGION"])["ARRIVAL_STATION_CD"] \
               .apply(lambda x: ", ".join(x.unique()[:3])) \
               .reset_index()

summary = summary.merge(examples, on=["TIME_OF_DAY", "HAUL", "ARRIVAL_REGION"])
summary.rename(columns={"ARRIVAL_STATION_CD": "Example Destinations"}, inplace=True)

# Step 6: Create a readable "Grouping" column
summary["Grouping"] = summary["TIME_OF_DAY"] + ", " + summary["HAUL"] + ", " + summary["ARRIVAL_REGION"]

# Step 7: Add notes (simple rule-based examples)
def add_notes(row):
    # Convert percentages from strings like "14.0%" to floats
    t1 = float(row["Tier1_%"].strip("%"))
    t2 = float(row["Tier2_%"].strip("%"))
    t3 = float(row["Tier3_%"].strip("%"))

    # Build a descriptive note tied directly to the numbers
    note = f"Tier1={t1}% (First minimal), Tier2={t2}% (Gold/Business significant), Tier3={t3}% (Club dominates)."

    # Add contextual interpretation
    if t1 < 1 and t3 > 75:
        note += " → No First Class; Club Lounge dominates."
    elif t1 >= 3 and t2 >= 12:
        note += " → Premium demand stronger; Business cabins significant, though Club Lounge dominates."
    elif t2 >= 14 and t3 >= 80:
        note += " → Balanced demand; Business and Club Lounge usage both strong."
    else:
        note += " → Mixed demand."

    return note

summary["Notes"] = summary.apply(add_notes, axis=1)


# Step 8: Final lookup table
final_table = summary[["Grouping", "Example Destinations", "Tier1_%", "Tier2_%", "Tier3_%", "Notes"]]
final_table
```

	Grouping	Example Destinati...				Notes
0	Afternoon, LONG, Asia	HND	4.0%	14.0%	82.0%	Tier1=4.0% (First minimal), Tier2=14.0
1	Afternoon, LONG, Middle East	DXB	4.0%	15.0%	82.0%	Tier1=4.0% (First minimal), Tier2=15.0
2	Afternoon, LONG, North America	LAX, ORD, JFK	4.0%	14.0%	82.0%	Tier1=4.0% (First minimal), Tier2=14.0
3	Afternoon, SHORT, Europe	FRA, VIE, CDG	0.0%	20.0%	80.0%	Tier1=0.0% (First minimal), Tier2=20.
4	Evening, LONG, Asia	HND	4.0%	14.0%	82.0%	Tier1=4.0% (First minimal), Tier2=14.0
5	Evening, LONG, Middle East	DXB	4.0%	14.0%	82.0%	Tier1=4.0% (First minimal), Tier2=14.0
6	Evening, LONG, North America	JFK, ORD, LAX	4.0%	14.0%	82.0%	Tier1=4.0% (First minimal), Tier2=14.0
7	Evening, SHORT, Europe	IST, FRA, VIE	0.0%	20.0%	80.0%	Tier1=0.0% (First minimal), Tier2=20.
8	Lunchtime, LONG, Asia	HND	4.0%	15.0%	82.0%	Tier1=4.0% (First minimal), Tier2=15.0
9	Lunchtime, LONG, Middle East	DXB	4.0%	14.0%	82.0%	Tier1=4.0% (First minimal), Tier2=14.0
10	Lunchtime, LONG, North America	ORD, JFK, DFW	4.0%	15.0%	82.0%	Tier1=4.0% (First minimal), Tier2=15.0
11	Lunchtime, SHORT, Europe	FRA, CDG, AMS	0.0%	20.0%	80.0%	Tier1=0.0% (First minimal), Tier2=20.
12	Morning, LONG, Asia	HND	4.0%	14.0%	82.0%	Tier1=4.0% (First minimal), Tier2=14.0
13	Morning, LONG, Middle East	DXB	3.0%	15.0%	82.0%	Tier1=3.0% (First minimal), Tier2=15.0
14	Morning, LONG, North America	LAX, JFK, ORD	4.0%	14.0%	82.0%	Tier1=4.0% (First minimal), Tier2=14.0
15	Morning, SHORT, Europe	IST, AMS, MUC	0.0%	20.0%	80.0%	Tier1=0.0% (First minimal), Tier2=20.

Rows: 16 Expand

```
final_table.to_csv("lookup_table.csv", index=False)
```